TU WIEN Informatics

Increasing Efficiency and Flexibility in Post-Quantum Cryptography

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Valerio Cini
Registration Number 11936028

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Matteo Maffei
Second advisor: Dr. Daniel Slamanig

The dissertation has been reviewed by:

Prof. Dennis Hofheinz	Prof. David J. Wu

Vienna, 13th October, 2023

Valerio Cini

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Declaration of Authorship

Valerio Cini

I hereby declare that I have written this Doctoral Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

Vienna, 13th October, 2023

Valerio Cini

Acknowledgements

I want to take a moment to express my gratitude to the amazing people who have been part of my Ph.D. journey.

First of all, to my supervisor, Daniel Slamanig. He introduced me to the world of research and patiently guided me through my Ph.D. I have always felt he was my strongest supporter by encouraging me to study any topic I was interested in and to expand my collaborations. I was very fortunate to have Daniel as supervisor, collaborator and teacher. I also want to acknowledge Matteo Maffei for supervising my graduate studies at TU Wien.

To Sebastian Ramacher and Christoph Striecks, your discussions and patience in answering my questions were incredibly helpful. The research meetings at Café Rudigerhof were something I'll miss. I also want to thank all my other colleagues at AIT - Thomas, Stephan, Paul, Luca, Stephan, Florian, and Julia.

A special thank goes to Giulio Malavolta for welcoming me for an internship at MPI Security and Privacy. The experience there was incredible, sparking numerous new research ideas and collaboration projects. I am also thankful to Russell W. F. Lai for hosting me at Aalto. I learned a lot during my time there. Additionally, I am grateful to Hoeteck Wee for hosting me in San Francisco and providing me with invaluable advices and research suggestions.

I also had the pleasure of collaborating with Erkan, Martin, Aravind, Hamza, Ivy, and Khanh - working with you was a real pleasure.

A special mention goes to Ahmadreza and Phillip, you made my time in Bochum much better than I could have hoped for. I also want to extend my thanks to everyone else I met at MPI, especially Behzad, Pedro, and Hendrick. Likewise to Gers, whose enthusiasm for biking and culinary skills added some fun moments to my stay in Bochum.

I am grateful to Prof. Hofheinz and Prof. Wu for accepting to be referees for this thesis, the thorough reviews, and the helpful comments.

I am very thankful to my family and old friends for their constant support.

And last, but definitely not least, I am in debt with Guendalina, for always being there and to keep putting up with me in spite of everything. Thank you.

Kurzfassung

Im Jahr 1994 entwickelte Peter Shor (FOCS 1994) einen Polynomialzeit-Quantenalgorithmus, mit dem die Sicherheit von kryptographischen Protokollen gebrochen werden kann, die auf der Härte des Faktorisierungsproblems oder der Berechnung von diskreten Logarithmen beruhen. Dies initierte die Untersuchung von Problemen, für die derzeit keine effizienten Quantenalgorithmen bekannt sind. Es entstand ein neuer Zweig der Kryptographie, die so genannte Post-Quanten-Kryptographie. Sie versucht, kryptographische Primitiven aus der vermuteten Schwierigkeit solcher Probleme zu konstruieren. Die Einführung von Post-Quanten-Kryptosystemen wird jedoch derzeit durch mindestens zwei Faktoren erschwert: 1) die derzeit eingesetzten Kryptosysteme sind schneller und/oder kompakter als Post-Quanten-Kryptosysteme, 2) in vielen modernen Szenarien werden kryptografische Protokolle umfangreichere Funktionen benötigen, als sie die derzeitigen Post-Quanten-Konstruktionen unterstützen. Beispielsweise existiert kein Post-Quanten-Kandidat für ein succinct non-interactive argument of knowledge (SNARK) mit Eigenschaften von (Prä-Quanten-)Konstruktionen, die auf bilinearen Paarungen basieren.

In dieser Arbeit machen wir Fortschritte auf beiden Seiten. Konkreter, können die Ergebnisse wie folgt zusammengefasst werden.

- Wir untersuchen, wie man asymmetrische Verschlüsselungsverfahren (PKEs) mit potenziell großen Korrektheitsfehlern in solche mit vernachlässigbaren Korrektheitsfehlern transformieren kann. Insbesondere zeigen wir, dass der direkte Produkt-Compiler von Dwork, Naor und Reingold (EUROCRYPT 2004) in Kombination mit einer Transformation von Hofheinz, Hövelmanns und Kiltz (TCC 2017) verwendet werden kann, um schwach sichere deterministische oder randomisierte PKEs generisch in CCA-sichere KEMs im sogenannten (Quanten-)Random Oracle Model (ROM) zu transformieren. Eine solche Transformation ist auf alle Kandidaten des NIST-Post-Quantem-Wettbewerbs, die auf Gittern und Codes mit nicht vernachlässigbarem Fehler basieren, anwendbar. Wir liefern eine umfassende Analyse und zeigen, dass sie die konkrete Effizienz einiger der codebasierten Kandidaten verbessert. Darüber hinaus zeigen wir, wie dieselben Ideen angewandt werden können, um einen ersten Ansatz für sichere Post-Quantem Bloom-Filter KEMs zu erhalten, ein Primitiv, das von Derler et al. (EUROCRYPT 2018) in Verbindung mit

punktierbaren KEMs eingeführt wurde. Dies liefert uns erstmals solche Verfahren basierend auf Gittern und Codes.

- Wir schlagen den ersten gitterbasierten SNARK vor, der gleichzeitig viele wünschenswerte Eigenschaften erfüllt: (i) er ist vorläufig post-quantum sicher, (ii) er ist öffentlich verifizierbar, (iii) Verifizieren benötigt logarithmische Zeit und (iv) er ist rein algebraisch und somit für eine effiziente rekursive Komposition geeignet. Das Herzstück dieser Konstruktion ist ein neues gitterbasiertes Vector Commitment Schema das das Öffnen zu multivariaten polyomialen Abbildungen konstanten Grades unterstützt. Die Sicherheit unserer Konstruktionen basiert auf einer neuen Familie von gitterbasierten Härteannahmen, die die Standardannahme der Short Integer Solution (SIS) auf natürliche Weise verallgemeinert.

- Aufbauend auf der obigen Konstruktion stellen wir einige weitere Ansätze zur Konstruktion effizienter gitterbasierter SNARKs vor. Insbesondere schlagen wir ein neues Commitment Schema vor, das auf verschwindenden Polynomen "vanishing Polynomials" basiert, einem Konzept das aus der algebraischen Geometrie stammt. Wir analysieren die Sicherheit eines solchen Commitment Schemas und zeigen, wie man die zusätzliche algebraische Struktur ausnutzen kann, um (i) das erste rekursiv faltende (d.h. Bulletproof-ähnliche) Protokoll für lineare Relationen mit poly-logarithmischer Verifier-Laufzeit und (ii) das erste gitterbasierte Linearzeit-Prover-Succinct-Argument für die Klasse NP im sogenannten Preprocessing-Modell zu erstellen.

Abstract

In 1994, Peter Shor (FOCS 1994) discovered a polynomial time quantum algorithm that can be used to break the security of protocols based on the hardness of factoring or computing discrete logarithms. This ignited the study of problems for which no quantum speed-ups are currently known, and a branch of cryptography, called post-quantum cryptography, started, trying to construct cryptographic primitives from the presumed hardness of such problems. However, the deployment of post-quantum cryptosystem is momentarily held back by at least two factors: 1) currently deployed cryptosystems are faster and/or smaller than post-quantum ones, 2) in many modern computing settings, richer functionalities are required from cryptographic protocols, than those that current post-quantum constructions support.

In this thesis we make progress on both sides. More concretely

1. We study the setting of generically transforming PKE schemes with potentially large correctness error to ones having negligible correctness error. In particular, we show that the direct product compiler by Dwork, Naor, and Reingold (EUROCRYPT 2004) can be used in combination with a transformation from Hofheinz, Hövelmanns, and Kiltz (TCC 2017) to generically transform weakly secure deterministic or randomized PKEs into CCA-secure KEMs in the (quantum) random oracle model. Such transformation applies to essentially all candidates to the NIST post-quantum competition based on lattices and codes with non-negligible error for which we provide an extensive analysis, showing that it improves the concrete efficiency of some of the code-based candidates. Moreover, we demonstrate how the same ideas can be applied to obtain a first approach towards post-quantum secure Bloom-Filter KEMs, a primitive introduced by Derler et al. (EUROCRYPT 2018) in connection with puncturable KEMs, generically from lattices and codes.

2. We propose the first lattice-based SNARK that simultaneously satisfies many desirable properties: (i) is tentatively post-quantum secure, (ii) is publicly-verifiable, (iii) has a logarithmic-time verifier and (iv) has a purely algebraic structure making it amenable to efficient recursive composition. At the heart of this construction is a new lattice-based vector commitment scheme supporting openings to constant-degree multivariate polynomial maps. The security of our constructions is based on

a new family of lattice-based computational assumptions which naturally generalises the standard Short Integer Solution (SIS) assumption.

3. Building on the above construction, we present some further approaches to constructing efficient lattice-based succinct arguments. In particular, we propose a new commitment scheme based on vanishing polynomials, a notion borrowed from algebraic geometry. We analyse the security of such a commitment scheme, and show how to take advantage of the additional algebraic structure to build (i) the first recursive folding (i.e. Bulletproofs-like) protocol for linear relations with poly-logarithmic verifier runtime, and (ii) the first lattice-based linear-time prover succinct argument for NP, in the preprocessing model.

List of Publications

[CRSS20] Valerio Cini, Sebastian Ramacher, Daniel Slamanig, and Christoph Striecks. CCA-Secure (Puncturable) KEMs from Encryption with Non-Negigible Decryption Errors. In Shiho Moriai and Huaxiong Wang, editors. *Advances in Cryptology - ASIACRYPT 2020, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 159-190, Daejeon, South Korea, December 7-11, 2020. Springer, Heidelberg, Germany.

[ACL+22] Martin R. Albrecht, Valerio Cini, Russel W. F. Lai, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable - (Extended Abstract). In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 102-132, Santa Barbara, CA, USA, August 15-18, 2022. Springer, Heidelberg, Germany.

[CLM23] Valerio Cini, Russel W. F. Lai, and Giulio Malavolta. Lattice-Based Succinct Arguments from Vanishing Polynomials - (Extended Abstract). (to appear in *Advances in Cryptology - CRYPTO 2023*)

Additional Publications

[CRS+21] Valerio Cini, Sebastian Ramacher, Daniel Slamanig, Christoph Striecks, and Erkan Tairi. Updatable signatures and message authentication codes. In Juan Garay, editor, *PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 12710 of *Lecture Notes in Computer Science*, pages 691-723, Virtual Event, May 10-13, 2021. Springer, Heidelberg, Germany.

[AC23] Hamza Abusalah and Valerio Cini. An incremental PoSW for general weight distributions. In *Advances in Cryptology - EUROCRYPT 2023, Part II*, Lecture Notes in Computer Science, pages 282-311. Springer, Heidelberg, Germany.

[CRS+23] Valerio Cini, Sebastian Ramacher, Daniel Slamanig, Christoph Striecks, and Erkan Tairi. (Inner-Product) Functional Encryption with Updatable Ciphertexts. (to appear in *Journal of Cryptology 2023*)

[CW23] Valerio Cini and Hoeteck Wee. ABE for Circuits with $\mathsf{poly}(\lambda)$-sized Keys from LWE. (to appear in *FOCS 2023*)

Contents

CHAPTER 1

Introduction

Most cryptographic primitives that play a crucial role in ensuring the confidentiality and authenticity of communications on the Internet and other networks rely on the mathematical hardness of problems related to factoring and discrete logarithms. In 1994, Peter Shor [Sho94] discovered polynomial time quantum algorithms that solve the hidden subgroup problem (HSP) for finite Abelian groups. Since both factoring and discrete logarithm are instance of HSPs in algebraic structures relevant to public-key cryptography, Shor's algorithm can be used to break the security of such protocols. Therefore, public-key cryptosystems that are currently used in practice will become insecure once sufficiently powerful quantum computers can be built. This changes will mainly interest public-key cryptography. Indeed, the only quantum speed-up known in the symmetric setting is the Grover's algorithm [Gro96] for search in unstructured databases. Such algorithm allows only a quadratic speed-up with respect to the brute-force search, and thus one can compensate it by increasing the key sizes. Moreover, since the run-time is also asymptotically optimal [BBBV97], (black-box) symmetric-key cryptography is post-quantum secure.

Not only currently used cryptography, like RSA, (EC)DH, (EC)DSA and related systems, would not stand the advent of quantum computers but, in general, it is preferable to have a wider set of assumptions on which one can build cryptographic primitives, so that even in case a major algorithmic breakthrough, primitives can be rapidly substituted by ones whose security has not been compromised. For this reasons, in the last decades numerous different assumptions have been extensively studied. The most well-established assumptions, that allow to instantiate different classes of cryptographic systems and that are considered post-quantum secure, are:

- Hash-based cryptography: Merkle's hash-tree signature scheme [Mer79],
- Code-based cryptography: McEliece public-key encryption scheme [McE78],

- Multivariate-quadratic-equations cryptography [Pat96],
- Isogeny-based cryptography: first efficient key exchange SIDH [JD11],
- Lattice-based cryptography: with the Ajtai-Dwork [AD97] being the first of various public-key constructions.

Lattice-based cryptography. Of the above listed class of assumption, lattice-based cryptography seems to provide the most promising general-purpose algorithms for public-key encryption/KEM and digital signature schemes [AASA+20]. Indeed, it provides quite efficient and parallelizable constructions, using mainly matrix and vector arithmetic as the basic operations, and allows us to replace essentially all of the currently endangered schemes. Moreover, lattice problems even allowed for the first time the construction of entire new classes of extremely powerful cryptographic tools. For example, Fully Homomorphic Encryption (FHE), a cryptographic primitive that allows arbitrary computation on encrypted data, was concived in 1978 by Rivest et al. [RAD+78] but remained an open problem for more than thirty years. In 2009 Craig Gentry [Gen09] provided a first instantiation based on ideal lattices, to which a number of constructions such as [vGHV10], [BV11], and [GSW13] followed. Similarly, the signature analogue of FHE, Fully Homomorphic Signature (FHS), has had somewhat practical instantiations via lattices only [GVW15b].

Furthermore, lattice-cryptography also provides strong security guarantees: in cryptography one is usually concerned with random instances of a given problem, whose (assumed) hardness implies the security of the system. However, in most cases, complexity theory only provides worst-case lower bounds, i.e., one is only able to prove that no algorithm is faster than the given bound for *all* instances of the problem. This does not imply that a random instance is hard: there might be problems where one single instance is out of reach with current algorithms, but where random ones are particularly simple. What one would require to prove the security of the scheme is therefore an average-case hardness result for some explicit (efficiently samplable) distribution. In cryptographic constructions based on worst-case hardness, such questions do not even arise. In lattice-based cryptography it has been shown, starting with the seminal work of Ajtai [Ajt96] and Micciancio and Regev [MR04] that there are reductions between worst-case lattice problems like approximate Shortest Vector Problem (approx-SVP), and average-case ones, like Short Integer Solution (SIS). This implies that cryptographic constructions which are secure based on the average-case hardness of SIS, actually reduce their security to the worst-case hardness of approx-SVP.

Towards practical deployment. If all the above considerations are true, a natural question that arises is why we haven't yet migrated to post quantum-secure schemes. The answer to this question is at least two-fold.

On the one hand efficiency: currently deployed cryptosystems are faster and/or smaller than post-quantum ones [NDR+19]. Therefore, before being able to utilize them in real-world scenarios, these new algorithms need further improvements and optimizations.

On the other hand confidence: to build trust in these assumptions, the community needs to make sure that cryptanalysts have taken time to search for attacks on such systems. Those cryptanalysts, in turn, need to gain familiarity with post-quantum cryptography and experience with post-quantum cryptanalysis [BBD09]: lattice-based cryptography uses a multitude of different computational hardness assumptions such as Learning with Errors (LWE), ring LWE [Reg05], SIS, ring SIS, their module variants, etc. Therefore, further research is required to establish the hardness of the most relevant of these problems in detail [BBGP16]. Further analysis has been triggered by the National Institute of Standards and Technology (NIST) due to their post-quantum competition (NIST PQC)[1] to standardize replacements for our current public-key cryptosystems in order to prepare for the eventuality that large-scale quantum computers become a reality. Started in 2017, the NIST PQC is now going through a 4th round of submission, with some candidates to be standardized already announced: three out of four of these candidates are lattice-based (Kyber, Dilithium, and FALCON).

Interest for a possible real-world deployment of such cryptosystems has been also recently shown by Cloudflare and Google who ran post-quantum TLS[2] experiments, to gain insights into the performance of these new constructions in real-world scenarios. Also the AWS Key Management Service now supports three new hybrid post-quantum key exchange algorithms for the Transport Layer Security protocol.[3]

Increasing demand for more flexible cryptography. Efficiency, however, is not the only criteria for real-world deployment. In many modern computing settings, richer functionality is required from cryptographic protocols: think, for example, of an application where data is collected by some organizations (e.g., hospitals), stored and processed on remote servers (e.g., the Cloud) and finally consumed by other users (e.g., medical researchers) on different devices. In these kind of scenarios, not only each party might need functionalities beyond those provided by standard primitives, but also, given the complexity of interactions and the different, concurrent needs of each party, stronger security guarantees are required (coming back to the previous example: is the Cloud trusted? If so to which level and by which parties?). Similar situations also often occur in the setting of the Internet of Things (IoT) and Edge computing. Another domain that is on the forefront of deploying advanced cryptography and has been a strong catalyst for it in the last few years is the field of cryptocurrencies and more generally distributed ledger technologies. Here, for instance, privacy-enhancing technologies and zero-knowledge proof systems used to achieve stronger anonymity guarantees have been extensively studied,

[1] https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization/Call-for-Proposals

[2] https://blog.cloudflare.com/the-tls-post-quantum-experiment/

[3] https://aws.amazon.com/blogs/security/round-2-post-quantum-tls-is-now-supported-in-aws-kms/

implemented, optimized and deployed. However, despite such efforts, currently various post-quantum primitives (e.g., succinct non-interactive arguments of knowledge) do not match the desirable features of (pre-quantum) constructions based on bilinear pairings.

Aim of this thesis. Given the discussion above, the aim of this thesis is to investigate post-quantum cryptography - and in particular lattice-based cryptography - with regard to the following two aspects:

- the design of novel cryptographic construction that improve efficiency aspects of previous instantiations, and
- the investigation whether the expressiveness offered by lattice assumptions can be exploited to add useful and novel features to (conventional) cryptographic primitives such that they provide the desired flexibility to be useful to modern communication scenarios.

1.1 Summary of Contributions

The main body of this thesis is composed of three chapters, each of them corresponding to one of the following publications:

- CCA-Secure (Puncturable) KEMs from Encryption with Non-Negligible Decryption Errors [CRSS20],
- Lattice-based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable [ACL+22],
- Lattice-based Succinct Arguments from Vanishing Polynomials [CLM23].

Now, we provide a detailed overview of the contributions made by each of these works, illustrating their relevance within the context presented earlier.

1.1.1 CCA-Secure (Puncturable) KEMs from Encryption with Non-Negligible Decryption Errors [CRSS20]

The standard security notion of encryption is security against chosen-ciphertext attacks (IND-CCA security), necessary in order to avoid malleability of ciphertexts and attacks in practical deployments of such schemes [Ble98]. There are various compilers to obtain such a strong security guarantees from schemes with weaker security. Among them the most widely used, especially for its use of the random oracle methodology, and thus better performance, is the Fujisaki-Okamoto (FO) transform [FO99]. Recently, Hofheinz, Hövelmanns, and Kiltz (HHK) [HHK17a] investigated different variants of the FO transform also in a setting where the underlying encryption scheme has non-perfect

correctness, but where decryption errors may occur with at most a negligible probability in the security parameter. This is interesting since many PKE schemes or KEMs based on conjectured quantum-safe assumptions and in particular assumptions on lattices and codes do not provide perfect correctness. Even worse, some of the candidates submitted to the NIST post-quantum competition (PQC) suffer from a *non-negligible* correctness error and so the FO transforms of HHK cannot be applied.

A natural question is therefore whether it is possible to construct a compiler that fits in the HHK framework but that can operate on schemes with non-negligible correctness error.

To this end, we started by revisiting the work of Dwork et al. [DNR04]. In this work, Dwork, Naor, and Reingold present a direct product compiler (which we dubbed $\mathsf{C}_{\mathsf{p,r}}$ and $\mathsf{C}_{\mathsf{p,d}}$ for randomized and deterministic PKEs, respectively). This compiler takes as input a PKE scheme $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ with non-negligible correctness error δ and outputs a PKE scheme $\Pi' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ with negligible correctness error δ'. The two compilers are quite easy to explain: the encryption algorithm Enc' works by encrypting the same message multiple times in parallel via Enc, under different public keys of the underlying PKE Π, which are generated at the key-generation phase, in case of a deterministic PKE, or using freshly sampled independent randomness, in case of a randomized PKE. The decryption algorithm Dec' works by decrypting each of the component of the ciphertext using the decryption algorithm Dec of the underlying PKE Π, and returning the message returned most often. In case two or more messages are tied, one of them is returned arbitrarily. We denote this last operation by maj.

Note that, as far as the deterministic direct product compiler $\mathsf{C}_{\mathsf{p,d}}$ is concerned, the correctness error can be improved by modifying the decryption: instead of relying on the maj operation, we can re-encrypt the plaintexts obtained during decryption with the respective keys and compare them to the original ciphertexts. Only if this check passes, the plaintext is returned. If this is done, then decryption fails with probability $\ell\delta^{\ell}$, where ℓ being the number of parallel repetitions, and thereby the number of parallel repetition necessary to achieve negligible correctness-error is reduced at the cost of a computational overhead during decryption. We denote this version of the deterministic direct product compiler by $\mathsf{C}^{\star}_{\mathsf{p,d}}$. Clearly this strategy is not possible when the PKE is randomized, and exactly the fact that one has to rely on a majority vote in case of a tie, makes the concrete efficiency of the direct product compiler slightly worse than what one could hope in the case of randomized PKE: even if the correct message has been returned, one has no way of checking that in case of a tie.

Since we were interested in concrete efficiency of the resulting scheme, we therefore explored an alternative route and investigated the possibility of starting from an $\mathsf{IND\text{-}CPA}$ secure PKE Π with non-negligible correctness error δ and introduce a variant of the transform T, introduced in the work of Hofheinz et. al. [HHK17a], to de-randomize a PKE, denoted $\mathsf{T}^{\star}$. The idea is that we compute ℓ independent encryptions of the same message M under the same public key pk using randomness $\mathsf{G}(M, i)$, $i \in [\ell]$, where G is a random oracle (RO). The resulting de-randomized PKE Π' has then correctness error

$\Pi'.\mathsf{Enc}(\mathsf{pk}, M)$	$\Pi'.\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$
for $i = 1, \ldots, \ell$ **do**	$\mathtt{res} \leftarrow \bot$, $\mathtt{check} \leftarrow \bot$
$\quad C_i := \Pi.\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M, i))$	**for** $i = 1, \ldots, \ell$ **do**
$C := (C_1, \ldots, C_\ell)$	$\quad \mathtt{res}[i] := \Pi.\mathsf{Dec}(\mathsf{sk}, C_i)$
return C	**for** $i \in [\ell]$ s.t. $\mathtt{res}[i] \neq \bot$ **do**
	$\quad$ **if** $\forall j \in [\ell] : \mathsf{ctxt}_j = \Pi.\mathsf{Enc}(\mathsf{pk}, \mathtt{res}[i]; \mathsf{G}(\mathtt{res}[i], j))$
	$\quad\quad \mathtt{check} \leftarrow i$
	if $\mathtt{check} \neq \bot$
	$\quad$ **return** $\mathtt{res}[\mathtt{check}]$
	return $\bot$

Figure 1.1: OW-PCA-secure scheme $\Pi' = \mathsf{T}^\star[\Pi, \mathsf{G}]$ with deterministic encryption and correctness error δ^ℓ from IND-CPA secure scheme Π with correctness error δ.

$\delta' := \ell\delta^\ell$, where ℓ is chosen in a way that $\ell\delta^\ell$ is negligible. A formal description of the transform $\mathsf{T}^\star$ is give in Figure 1.1. To the resulting PKE Π' we can then directly apply the transformation $\mathsf{U}^{\not\perp}$ from [HHK17a] to obtain an IND-CCA secure KEM KEM with negligible correctness error in the (Q)ROM.

Note that as we directly integrated the product compiler into the T transform from [HHK17a], the correctness of the message can be "checked" via the de-randomization. Hence, we could get rid of the majority vote in the direct product compiler. With this change the analysis of the concrete choice of ℓ became simpler and, more importantly, allowed us to choose smaller ℓ than in the black-box use of the compiler.

Next, we analyzed the transform both in the ROM and QROM, giving a tight reduction in the ROM, and compared it to a generic application of the direct product compiler. Since our transform naturally fits into the modular framework of HHK [HHK17a], by applying the $\mathsf{U}^{\not\perp}$ transform, gives rise to an IND-CCA-secure KEM. For the analysis in the QROM, we followed the work of Bindel et al. [BHH+19]. We showed that the $\mathsf{T}^\star$ transform also fits into their framework. Hence, given the additional injectivity assumption, we also obtained a tight proof for $\mathsf{U}^{\not\perp}$. But even if this assumption does not hold, the non-tight proofs of Jiang et al. [JZM19] and Hövelmanns et al. [HKSU20] remain applicable. Compared to the analysis of the T transform that is used in the modular frameworks, our reductions lose a factor of ℓ, i.e., the number of parallel ciphertexts required to reach a negligible correctness error, in the ROM and a factor of ℓ^2 in the QROM. For concrete schemes, this number is small (e.g., ≤ 5) and, thus, does not impose a significant loss. An overview of the transformations and how our transform fits into the modular frameworks is given in Figure 1.2 (ROM) and Figure 1.3 (QROM).

After analyzing how our new compiler fits in the HHK framework, we evaluated the concrete efficiency of the $\mathsf{T}^\star$ transform based on its application to code- and lattice-based second-round candidates in the NIST PQC. To do so, we focused on schemes that offered both an IND-CPA secure version with non-negligible correctness error and

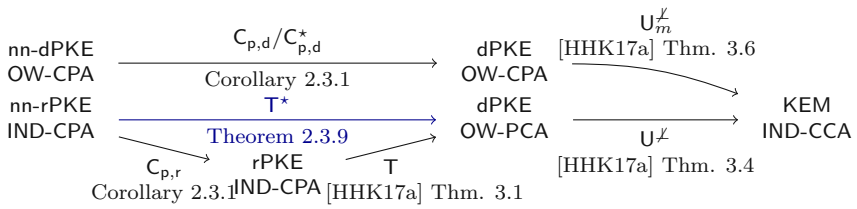

Figure 1.2: Overview of the transformations in the ROM with the results related to $\mathsf{T}^{\star}$ highlighted in blue. rPKE denotes a randomized PKE. dPKE denotes a deterministic PKE. The prefix nn indicates encryption schemes with non-negligible correctness error.

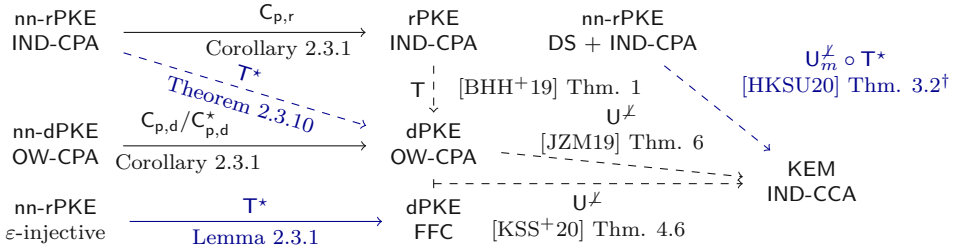

Figure 1.3: Overview of the transformations in the QROM using the notation from Figure 2.1. A dashed arrow denotes a non-tight reduction. DS denotes disjoint simulatability.
†: Obtained by applying the modifications from Theorems 2.3.9 and 2.3.10 to [HKSU20, Thm 3.2].

IND-CCA variant with negligible correctness error. We compared how the application of our transform to the IND-CPA variant performs against the IND-CCA version. For code-based schemes (an overview can be found in Table 1.1) such as ROLLO we can observe improvements in the combined size of public keys and ciphertexts, a metric important when the primitive is used in protocols such as TLS, as well as its runtime efficiency. We also argued the ease of implementing our so-obtained schemes which can rely on simpler decoders. For lattice-based constructions, we found that the use of the transform results in an increase in the sum of ciphertext and public-key size of 30% even in the best case scenario, i.e., for an IND-CPA version of the KEM Round5 [GZB+19].

In addition to the context of conventional PKE schemes and KEMs, we explored our approach further. Specifically, we focused on a class of KEMs known as puncturable KEMs, which have recently gained attention, particularly in the context of full forward-secrecy for zero round-trip time (0-RTT) key-exchange (KE) protocols. These puncturable KEMs [GM15, GHJL17, DJSS18, SSS+20] include Bloom Filter KEMs (BFKEMs) [DJSS18, DGJ+21]. BFKEMs are CCA-secure KEMs that possess an inherent non-negligible correctness error. However, the source of this non-negligible correctness error can be traced back to the Bloom filter layer whereas the underlying IBE scheme, specifically the Boneh-Franklin [BF01] in the instantiation of [DJSS18], is required to

Table 1.1: Sizes (in bytes) and runtimes (in ms and millions of cycles for BIKE), where O denotes the transformed scheme. The LEDAcrypt instances with postfix NN refer to those with non-negligible DFR. Runtimes are taken from the respective submission documents and are only intra-scheme comparable.

KEM	δ	pk	ctxt	$\sum$	KGen	Encaps	Decaps
O[ROLLO-I-L1,5]	$2^{-147.7}$	**465**	2325	**2790**	**0.10**	**0.02**/0.10	**0.26**/1.30
ROLLO-II-L1	2^{-128}	1546	1674	3220	0.69	0.08	0.53
O[ROLLO-I-L3,4]	2^{-126}	**590**	2360	**2950**	**0.13**	**0.02**/**0.08**	**0.42**/1.68
ROLLO-II-L3	2^{-128}	2020	2148	4168	0.83	0.09	0.69
O[ROLLO-I-L5,4]	2^{-166}	**947**	7576	8523	**0.20**	**0.03**/0.12	**0.78**/3.12
ROLLO-II-L5	2^{-128}	2493	2621	5114	0.79	0.10	0.84
O[BIKE-2-L1,3]	$2^{-145.4}$	**10163**	30489	40652	**4.79**	**0.14**/0.42	**3.29**/9.88
BIKE-2-CCA-L1	2^{-128}	11779	12035	23814	6.32	0.20	4.12
O[LEDAcrypt-L5-NN,2]	2^{-127}	22272	22272	44544	5.04	**0.14**/**0.29**	**1.55**/3.11
LEDAcrypt-L5	2^{-128}	19040	19040	38080	4.25	0.84	2.28

provide perfect correctness. As a consequence, since no current post-quantum IBEs exihibits perfect correctness, there were no known instantiation of post-quantum BFKEMs. In this work we made progress in this direction and showed that, using the ideas we previously introduced, it is possible to construct BFKEMs generically from any IBE, even from IBEs with (non-)negligible correctness error. Therefore, this allows BFKEMs to be instantiated from lattice- and code-based IBE, and provided the first candidates for post-quantum CCA-secure BFKEMs.

We note that our work has been done while the second round of the NIST PQC was still ongoing. In the meantime, the third-round candidates[4] and finalists[5] have been announced by NIST for standardization. From the schemes that are suitable for our compilers, BIKE [ABB+19] and FrodoKEM [NAB+19] were advanced to the third round as alternate candidates in the competition, and BIKE [ABB+19] also reached the fourth round of the competition. Moreover, we concretely analyze the submissions to the second round and want to note that meanwhile there are additional results on the cryptanalysis of some relevant second round schemes, e.g., for ROLLO in [BBC+20] as well as for LEDAcrypt in [APRS20]. These results might require a change in the parameters compared to the versions that we used in this work.

[4] https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement
[5] https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4

1.1.2 Lattice-based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable [ACL+22]

A succinct non-interactive argument of knowledge (SNARK) [Kil92, Mic94] allows a prover to convince a verifier that they know a witness to an NP statement. The succinctness property demands that the size of the proof and, after preprocessing, the work of the verifier are sublinear in (and ideally independent of) the time needed to check the validity of the witness. Over the last decade, SNARKs have witnessed a significant rise in their efficiency and applicability [Lip12, PHGR13, BCG+13, GGPR13, BCTV14b, Gro16]. More recently, SNARKs have found their way into real-world systems in the context of blockchain-based cryptocurrencies [BCG+14, KMS+16, BGH19, BMRS20, BDFG21a].

The most efficient and feature-rich SNARKs are constructed over bilinear groups (e.g., [Gro16]) with a trusted setup. Typically, a pairing-based SNARK proof consists of only a small constant number of base group elements and is also publicly verifiable. Furthermore, offline preprocessing can often be performed, such that the online verification time is sublinear in the size of the statement being proved and the corresponding witness. Moreover, pairing-based SNARKs are favourable because of their algebraic properties that are known to enable proof batching [LMR19, BMM+21] and efficient recursive composition [BCTV14a]. However, due to their reliance on the hardness of problems related to discrete logarithms, pairing-based SNARKs are not sound against a cheating quantum prover.

So far, known lattice-based schemes suffer from (at least) one of the following limitations:

- They require the verifier to hold a secret information that should not be made available to the prover, i.e. they are in the designated-verifier setting [BISW17, BISW18, GMNO18].
- They have a non-succinct verifier, whose runtime is at least linear in the size of the relation [BLNS20, AL21, ACK21].

In this work, we make progress in this direction by providing the first SNARK construction that satisfies the following properties at the same time: (plausibly) post-quantum (lattice-based construction), publicly verifiable, pre-processing (fast verifier), and completely algebraic (hence friendly to recursive composition)

Our construction qualitatively matches pairing-based SNARKs, and in addition, it is tentatively post-quantum secure. The soundness of our scheme is based on new lattice-based (knowledge) assumptions. The introduction of new knowledge assumptions is, to some extent, necessary: The work of Gentry and Wichs [GW11] shows that the adaptive soundness of any SNARK cannot be based on falsifiable assumptions in a black-box manner.

In order to do so, we take a new route to construct a SNARKs for NP: recall that satisfiability of a system of degree d equations is a NP complete language for any $d \geq 2$.

We show that there is a simple compiler to obtain a SNARK for NP from an extractable and succinct vector commitment supporting opening to degree $d \geq 2$ polynomial maps.

A vector commitment (VC) is a cryptographic primitive that allows a committer to commit to a vector of w values $\mathbf{x} \coloneqq (x_1, \ldots, x_w) \in \mathbb{Z}^w$ and then reveal selected portions of the input vector, or more generically a function $f : \mathbb{Z}^w \to \mathbb{Z}^t$ over the input vector, along with a proof π that can be publicly verified.

The standard security requirement for such primitive is called evaluation binding.

Evaluation Binding. It is computationally infeasible to produce a commitment c and two opening proofs, π and π', for the same polynomial map f, but different image values $\mathbf{y} \neq \mathbf{y}'$.

A stronger security notion is extractability.

Extractability. To produce a commitment c and a proof that the image of a polynomial map f at the committed vector is $\mathbf{y}$, one must know a preimage $\mathbf{x}$ such that c is a commitment of $\mathbf{x}$ and $f(\mathbf{x}) = \mathbf{y}$.

As far as efficiency of the construction is concerned, we will consider the following property.

Succinctness. The size of the commitment c and opening proof π for a polynomial map $f : \mathbb{Z}^w \to \mathbb{Z}^t$ is upper-bounded by a fix polynomial in $\mathsf{poly}(\lambda, \log w, \log t)$.[6]

In order to construct a SNARK for NP, it suffices to build an extractable and succinct vector commitment supporting opening to degree-2 polynomial. Indeed, given such a primitive, exploiting the fact that satisfiability of degree d equations is an NP-complete language for any $d \geq 2$, we can compile the VC into a SNARK as follows: The (SNARK) prover simply commits to the root $\mathbf{x}$ of the system $(f, \mathbf{y})$ and immediately produces an opening proof for $(f, \mathbf{y})$. The (SNARK) verifier simply runs the verification algorithm of the VC scheme. Succinctness requirements and knowledge soundness of the SNARK are derived respectively from the shortness of commitments and opening proofs of the VC, and from extractability of the VC. In this way, we have reduced the task of constructing a lattice-based SNARK to that of constructing a lattice-based VC with the above stated properties.

We make progress in this direction and construct a lattice-based VC supporting opening to constant-degree polynomial maps. In doing so, we develop a *framework for translating pairing-based constructions to the lattice world*. In order to prove such translated construction secure, we also map to the lattice-world the group assumptions under which such pairing constructions are proven secure.

[6] In Chapter 3 we will use more fine-grained definitions to distinguish different efficiency requirements.

To explain the basic ideas behind such lattice constructions and the assumptions under which we attempt to prove them secure, we describe now a vector commitment for linear functions (in w variables) that stems from applying our translation technique to pairing-based VC with openings to linear function adapted from [CF13, LRY16, LM19].

- Public Parameters: They consist of matrix and vectors $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{v} \leftarrow (\mathbb{Z}_q^{\times})^w$, $\mathbf{t} \leftarrow \mathbb{Z}_q^n$ and short preimages $\mathbf{u}_{i,j} \leftarrow \mathbf{A}^{-1}\left(\mathbf{t} \cdot \frac{v_i}{v_j}\right)$[7] for all $i, j \in [w], i \neq j$.
- Commitment: The commitment to some input value $\mathbf{x}$ will be produced by computing
$$c \coloneqq \langle \mathbf{v}, \mathbf{x} \rangle = \sum_{i \in [w]} v_i \cdot x_i$$
- Opening: The proof π for a function f such that $f(\mathbf{X}) = \sum_{i \in [w]} f_i \cdot X_i$ is a short vector $\mathbf{u}$ given by
$$\mathbf{u} \coloneqq \sum_{i,j \in [w], i \neq j} f_j \cdot x_i \cdot \mathbf{u}_{i,j}.$$
- Offline Pre-Computation: During the offline phase, the verifier, on input f and the public parameters, can compute the following value
$$\hat{f} \coloneqq \sum_{j \in [w]} f_j \cdot v_j^{-1}.$$
- Online Verification: The verifier, on receiving $\pi = \mathbf{u}$, checks that
  - $\mathbf{A} \cdot \mathbf{u} = \mathbf{t} \cdot (\hat{f} \cdot c - y) \bmod q$, and
  - $\mathbf{u}$ is short.

One can see that

$$\begin{aligned}
\hat{f} \cdot c &= \left(\sum_{j \in [w]} f_j \cdot v_j^{-1}\right) \cdot \left(\sum_{i \in [w]} v_i \cdot x_i\right) \\
&= \sum_{i,j \in [w]} f_j \cdot x_i \cdot \frac{v_i}{v_j} \\
&= \sum_{\substack{i,j \in [w] \\ i=j}} f_j \cdot x_i \cdot \frac{v_i}{v_j} + \sum_{\substack{i,j \in [w] \\ i \neq j}} f_j \cdot x_i \cdot \frac{v_i}{v_j} \\
&= f(\mathbf{x}) + \sum_{\substack{i,j \in [w] \\ i \neq j}} f_j \cdot x_i \cdot \frac{v_i}{v_j}.
\end{aligned}$$

[7] $\mathbf{A}^{-1}\left(\mathbf{t} \cdot \frac{v_i}{v_j}\right)$ denotes a short preimage $\mathbf{u}_{i,j}$ satisfying $\mathbf{A} \cdot \mathbf{u}_{i,j} = \mathbf{t} \cdot \frac{v_i}{v_j} \bmod q$

Each term in the summation over $i, j \in [w]$ has a multiplicative factor of the form $\frac{v_i}{v_j} \neq 1$, as $i \neq j$. We can interpret the result as follows: by multiplying $\hat{f}$ and c, we get a Laurent polynomial (in $v_1, \ldots, v_w$) where the constant coefficient is exactly $f(\mathbf{x})$.

Therefore, if the claimed image value y indeed equals $f(\mathbf{x})$, then

$$\hat{f} \cdot c - y = \sum_{\substack{i,j \in [w] \\ i \neq j}} f_j \cdot x_i \cdot \frac{v_i}{v_j}$$

is a Laurent polynomial with constant coefficient equal to 0.

It follows that during verification

$$\begin{aligned}
\mathbf{A} \cdot \mathbf{u} &= \mathbf{A} \cdot \left(\sum_{\substack{i,j \in [w] \\ i \neq j}} f_j \cdot x_i \cdot \mathbf{u}_{i,j} \right) \\
&= \sum_{\substack{i,j \in [w] \\ i \neq j}} f_j \cdot x_i \cdot \mathbf{A} \cdot \mathbf{u}_{i,j} \\
&= \sum_{\substack{i,j \in [w] \\ i \neq j}} f_j \cdot x_i \cdot \mathbf{A} \cdot \mathbf{A}^{-1} \left(\mathbf{t} \cdot \frac{v_i}{v_j} \right) \\
&= \mathbf{t} \cdot \left(\sum_{\substack{i,j \in [w] \\ i \neq j}} f_j \cdot x_i \cdot \frac{v_i}{v_j} \right) \\
&= \mathbf{t} \cdot (\hat{f} \cdot c - y),
\end{aligned}$$

as required, and

$$\mathbf{u} = \sum_{\substack{i,j \in [w] \\ i \neq j}} f_j \cdot x_i \cdot \mathbf{u}_{i,j},$$

is somewhat short as it is a linear combination of short preimages ($\mathbf{u}_{i,j}$'s), with coefficients which are itself small (f_j's and x_i's) if we allow the prover to commit to only short vectors $\mathbf{x}$ and open to functions f with short representation.

Our translation techniques (and consequently security of the resulting cryptographic schemes) rely on a new family of assumptions that we refer to as the *k-Ring-Inhomogenous Short Integer Solution* (or *k-R*-ISIS for short) assumptions.[8] Roughly, the *k-R*-ISIS assumption (parametrized by a set of monomial $\mathcal{G}$ and a target monomial g^*) says that it

[8] in this introduction we set $R = \mathbb{Z}$

is hard to find a (short multiple of a) short preimage $\mathbf{u}_{g^*}$ satisfying $\mathbf{A}\cdot\mathbf{u}_{g^*} = \mathbf{t}\cdot g^*(\mathbf{v}) \bmod q$, where g^* is a target Laurent monomial and $\mathbf{v}$ is a random point, given short preimages under $\mathbf{A}$ of other Laurent monomials $\mathcal{G}$ evaluated on the same random point.

In the concrete case of the scheme described above, the set of Laurent monomials $\mathcal{G}$ for which preimages are given is

$$\{g_{i,j}\}_{i,j\in[w],i\neq j},$$

where $g_{i,j}(\mathbf{v}) = v_i/v_j$, and the target monomial g^* is simply the constant monomial 1. Indeed, suppose one has two accepting openings $\mathbf{u}, \mathbf{u}'$ for the same function f but different claimed output values $y \neq y'$, i.e.

$$\mathbf{A}\cdot\mathbf{u} = \mathbf{t}\cdot(\hat{f}\cdot c - y) \bmod q$$
$$\mathbf{A}\cdot\mathbf{u}' = \mathbf{t}\cdot(\hat{f}\cdot c - y') \bmod q,$$

with both $\mathbf{u}$ and $\mathbf{u}'$ short. Subtracting both equations, one obtains

$$\mathbf{A}\cdot(\mathbf{u}-\mathbf{u}') = \mathbf{t}\cdot(y'-y) \bmod q,$$

where $y'-y$ is a short multiple of the constant monomial 1.[9] That is, $\mathbf{u}^* \coloneqq (\mathbf{u}-\mathbf{u}')$ is a short preimage of a short multiple of the target monomial $g^*(\mathbf{v}) = 1$.

In this way we obtain a VC satisfying evaluation binding. There are still two properties left to achieve: i) upgrade security to achieve extractability, and ii) achieve output-succinct openings, i.e., to have opening scaling logarithmic with the output dimension t (and not linearly as one would get by simply concatenating opening proofs for each output entry).

To upgrade the security of such a scheme and obtain extractability, we propose a knowledge variant of the k-R-ISIS assumption. Recall that the introduction of new knowledge assumptions is, to some extent, necessary: the work of Gentry and Wichs [GW11] shows that the soundness of any SNARK cannot be based on falsifiable assumptions in a black-box manner.

For concreteness, we will use the following member of the knowledge k-R-ISIS assumption family:

> *Let* $\mathbf{B}\leftarrow\mathbb{Z}^{n\times m}$, $\mathbf{v}\leftarrow\mathbb{Z}_q^w$, *and* $\mathbf{t}\leftarrow\mathbb{Z}_q^n$ *be random matrix and vectors. If there exists an efficient algorithm* $\mathcal{A}$ *which, given short vectors* $\mathbf{u}_i'$ *satisfying* $\mathbf{B}\cdot\mathbf{u}_i' = \mathbf{t}\cdot v_i \bmod q$ *for all* $i\in[w]$, *produces* $(c,\mathbf{u}')$ *such that* $\mathbf{u}'$ *is a short vector satisfying* $\mathbf{B}\cdot\mathbf{u}' = \mathbf{t}\cdot c \bmod q$, *then there exists an efficient extractor* $\mathcal{E}_{\mathcal{A}}$ *which extracts a short vector* $\mathbf{x}\in\mathbb{Z}^w$ *such that* $\langle\mathbf{v},\mathbf{x}\rangle = c \bmod q$.

[9] Recall that both $\mathbf{x}$ and the coefficients of f are required to be short, and therefore also the claimed output values y's are forced to be so.

Equipped with this k-R-ISIS of knowledge assumption, we can upgrade our VC construction to achieve extractability as follows. First, we let the public parameters to additionally include $(\mathbf{B}, (\mathbf{u}'_i)_{i\in[w]}, \mathbf{t})$. Next, we let the committer also include $\mathbf{u}' = \sum_{i\in[w]} x_i \cdot \mathbf{u}'_i$ in an opening proof. Finally, we let the verifier additionally check that $\mathbf{u}'$ is short and $\mathbf{B} \cdot \mathbf{u}' = \mathbf{t} \cdot c \bmod q$.

To see why the modified scheme is extractable, suppose an adversary is able to produce a commitment c and a valid opening proof for (f, y). By the k-R-ISIS of knowledge assumption, we can extract a short vector $\hat{\mathbf{x}} \in \mathbb{Z}^w$ such that $\langle \mathbf{v}, \hat{\mathbf{x}} \rangle = c \bmod q$. Now, if $f(\hat{\mathbf{x}}) = y' \neq y$, we can use the extracted $\hat{\mathbf{x}}$ to compute a valid opening proof for (f, y'). However, being able to produce valid opening proofs for both (f, y) and (f, y') with $y \neq y'$ violates the evaluation binding property. We therefore conclude that $f(\hat{\mathbf{x}}) = y$.

To achieve output-succinctness we use a SIS instance to aggregate the opening proofs of each of the output entries. Specifically, the coefficients $\mathbf{h} = (h_i)_{i\in[t]} \in \mathbb{Z}$ that we use to aggregate opening proofs are given by an instance of the SIS problem over $\mathbb{Z}_p$ (taking smallest $\mathbb{Z}$-representatives of $\mathbb{Z}_p$ elements) sampled as part of the public parameters, where p is chosen such that the SIS assumption is believed to hold over $\mathbb{Z}_p$ while p is small relative to q.

To see why extractability still holds, suppose an adversary is able to produce a commitment c and a valid opening proof for (f, y) where $f = \sum_{i\in[t]} h_i \cdot f_i$ and $y = \sum_{i\in[t]} h_i \cdot y_i$. By our previous argument, we can extract $\mathbf{x}$ satisfying $f(\mathbf{x}) = y$. Suppose it is not the case that $f_i(\mathbf{x}) = y_i$ for all $i \in [t]$, then $(f_i(\mathbf{x}) - y_i)_{i\in[t]}$ is a short non-zero vector satisfying $\sum_{i\in[t]} h_i \cdot (f_i(\mathbf{x}) - y_i) = 0$ over $\mathbb{Z}$, which implies $\sum_{i\in[t]} h_i \cdot (f_i(\mathbf{x}) - y_i) = 0 \bmod p$, breaking the SIS assumption over $\mathbb{Z}_p$.

We have so far shown how to construct a lattice-based extractable VC for linear functions with succinct commitments and openings. It remains to show how to generalize this construction to support bounded-degree polynomial maps. This can be done exploiting the fact that we are now working over rings (and not groups anymore, as it was the case for pairing-based constructions). Indeed, we notice that each degree-d monomial $\mathbf{x}^{\mathbf{e}} = \prod_{i\in[w]} x_i^{e_i}$ is encoded in c^d as (a factor of) the coefficient of $\mathbf{v}^{\mathbf{e}} = \prod_{i\in[w]} v_i^{e_i}$. Each such monomial has a natural complement $\mathbf{v}^{-\mathbf{e}}$ satisfying $(\mathbf{v}^{\mathbf{e}}) \cdot (\mathbf{v}^{-\mathbf{e}}) = 1$. Using these facts, one can modify the definition of $\hat{f}$ appropriately, so that $\hat{f} \cdot c^d$ is a Laurent polynomial with constant coefficient $f(\mathbf{x})$ as before. Plugging in such a modification in the scheme described above yields a lattice-based extractable VC for bounded-degree polynomial maps as claimed. Notice that this time the evaluation binding property will be based on another appropriate member of the k-R-ISIS assumption family.

The SNARK obtained by compiling the lattice-based extractable VC supports proving the satisfiability of polynomial maps over $\mathbb{Z}$ (and more generally over any appropriate ring $\mathcal{R}$) by bounded-norm solutions, a language which directly captures those statements which naturally arise in lattice-based cryptographic constructions. We highlight two native applications of our SNARK. The first application is the recursive composition of our SNARK, which refers to the process of using the SNARK to prove knowledge

of another SNARK proof and the satisfiability of a polynomial map. This is natively supported because the verification algorithm of our SNARK construction is itself checking the satisfiability of certain algebraic relations over $\mathcal{R}$ by a bounded-norm solution. Recursive composition of SNARKs is a general purpose technique for aggregating proofs or proving complex statements in a piece-by-piece fashion. The technique is also useful for constructing incremental verifiable computation [Val08] and verifiable delay functions [BBBF18, Gro21].

The second application is the aggregation of GPV signatures [GPV08]. While it is folklore that any signatures can be aggregated by a SNARK for an NP-complete language, we stress that the GPV verification algorithm, again, checks the satisfiability of certain algebraic relations over $\mathbb{Z}$ by a bounded-norm solution which our SNARK natively supports. Apart from obtaining short aggregated GPV signatures, in the setting where a set of n signers are signing a common message at a time, the verification of the aggregated signatures could be preprocessed, resulting in an online verification time *sublinear* in n. As a bonus result on GPV signatures, we further show how to construct lattice-based adaptor signatures [AEE$^+$21] based on the GPV paradigm. Combining the two results, we obtain the first aggregatable adaptor signature.

Moreover, as a contribution of independent interest, we show that our VC satisfies a strong notion of binding known as *collapsing* (as an ordinary commitment, not with respect to functional openings), a recently introduced security notion in the quantum setting [Unr16]. For this, we introduce a new technique of embedding NTRU ciphertexts into the public parameters of our VC. To the best of our knowledge, this is the first VC not based on Merkle trees that is shown to satisfy such a notion.

1.1.3 Lattice-based Succinct Arguments from Vanishing Polynomials [CLM23]

As we have seen, a promising approach for constructing efficient SNARKs is to leverage the algebraic structure offered by computational problems in lattice-based cryptography [BISW17, BISW18, GMNO18, BLNS20, AL21, ACK21, ACL$^+$22].

At the same time, inspite of the recent progress presented in the previous section, lattice-based SNARKs are still somewhat limited compared to competing approaches. In particular, known (publicly-verifiable) lattice-based schemes are constructed following two different paradigms, each with some specific drawbacks:

- Bulletproofs-like arguments: They have a non-succinct verifier, whose runtime is at least linear in the size of the relation [BLNS20, AL21, ACK21].
- Knowledge-based arguments: They have a slow prover runtime, i.e., quartic [ACL$^+$22] in the size of the relation.

Let us briefly recall both approaches and analyze the source of their drawbacks.

Approach I: Folding Protocols. (Lattice-based) Bulletproofs [BLNS20, AL21, ACK21] are interactive arguments with quasi-linear time prover, and can be made non-interactive using the Fiat-Shamir transform in the random oracle model. They are based on the technique of iteratively "folding" the relation into a smaller one until a trivial relation is derived. Recall that in Bulletproofs the prover wants to convince the verifier that they know a short vector $\mathbf{x}$ satisfying

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0.$$

Let $(\mathbf{M}, \mathbf{x}, \mathbf{y}) = (\mathbf{M}^{(0)}, \mathbf{x}^{(0)}, \mathbf{y}^{(0)})$. The protocol consists of $\ell + 1$ rounds, where in the i-th round the two parties "fold" the relation represented by $(\mathbf{M}^{(i)}, \mathbf{y}^{(i)})$ into another represented by $(\mathbf{M}^{(i+1)}, \mathbf{y}^{(i+1)})$ where the dimension of $\mathbf{M}^{(i+1)}$ is half that of $\mathbf{M}^{(i)}$. Correspondingly, the prover folds its witness $\mathbf{x}^{(i)}$ into $\mathbf{x}^{(i+1)}$. After ℓ such folding steps, a constant-size relation $(\mathbf{M}^{(\ell)}, \mathbf{y}^{(\ell)})$ is reached and the prover simply sends the satisfying witness $\mathbf{x}^{(\ell)}$ over to the verifier.

In more detail, for $0 \leq i < \ell$, the i-th of the first ℓ rounds of the protocol proceeds as follows. The parties split $\mathbf{M}^{(i)}$ into two halves as $\mathbf{M}^{(i)} = (\mathbf{M}_L^{(i)}, \mathbf{M}_R^{(i)})$ and the prover splits $\mathbf{x}^{(i)} = (\mathbf{x}_L^{(i)}, \mathbf{x}_R^{(i)})$. The prover sends the cross terms

$$\mathbf{y}_{LR}^{(i)} = \left\langle \mathbf{M}_L^{(i)}, \mathbf{x}_R^{(i)} \right\rangle \bmod q \qquad \text{and} \qquad \mathbf{y}_{RL}^{(i)} = \left\langle \mathbf{M}_R^{(i)}, \mathbf{x}_L^{(i)} \right\rangle \bmod q.$$

The verifier sends a random challenge $r_i \leftarrow S$ sampled from some challenge set S. Both parties fold $(\mathbf{M}^{(i)}, \mathbf{y}^{(i)})$ into

$$(\mathbf{M}^{(i+1)}, \mathbf{y}^{(i+1)}) \coloneqq (\mathbf{M}_L^{(i)} + \mathbf{M}_R^{(i)} \cdot r_i^{-1}, \mathbf{y}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{y}^{(i)} + \mathbf{y}_{LR}^{(i)} \cdot r_i) \bmod q, \tag{1.1}$$

and the prover folds $\mathbf{x}$ into $\mathbf{x}^{(i+1)} = \mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i$. At the ℓ-th (i.e. last) round, the prover simply sends $\mathbf{x}^{(\ell)}$ and the verifier checks that $\mathbf{x}^{(\ell)}$ is short and satisfies $\left\langle \mathbf{M}^{(\ell)}, \mathbf{x}^{(\ell)} \right\rangle = \mathbf{y}^{(\ell)} \bmod q$.

It can been shown [BLNS20, AL21, ACK21, AF22] that the protocol satisfies knowledge soundness, and furthermore it is easy to see that the prover runs in time quasi-linear in the length of the witness. However, a major drawback of this approach is that the verifier computation is also quasi-linear for general linear relations $\mathbf{M}$, and it cannot be preprocessed due to the interactive nature of the scheme.

Approach II: Pre-Processing (Knowledge-Based) Protocols. This is the approach that we have introduced in the previous subsection and initiated in [ACL+22], and consists in compiling an extractable VC for degree $d \geq 2$ polynomial maps with short commitment and openings into a SNARK. The fact that this approach, as far as it has been presented in the previous section, leads to slow prover runtime, can be already seen in the construction of the VC for linear functions: the public parameters include preimages $\mathbf{u}_{i,j} \leftarrow \mathbf{A}^{-1}(\mathbf{t} \cdot \frac{v_i}{v_j})$ for $i, j \in [w], i \neq j$, where w is the length of the vectors that one commits to. That is, the number of preimages is $O(w^2)$, so already reading the

public parameters takes quadratic time. More generally, when generalizing this approach to support degree-d polynomial maps, the public parameters will have to include all ratios of (different) degree-d monomials. Since the number of such ratios is $O(w^{2d})$, i.e., exponential in the degree d, this forced us to only be able to support constant-degree polynomial maps.

Key Observation. Starting from [ACL+22], while trying to improve the prover runtime of our VC scheme, and consequently that of the SNARK, we made the following observation: if we allow the vector $\mathbf{v}$, over which the Laurent monomials g from the k-R-ISIS assumption are evaluated, to be structured and not uniformly random over $(\mathbb{Z}_q^\times)^w$ as before, then it is possible to reduce the size of the public parameters. For example, if we have

$$\mathbf{v} = (v, v^2, \ldots, v^w)^\intercal \in (\mathbb{Z}_q^\times)^w,$$

for some uniform random $v \in \mathbb{Z}_q^\times$, then $\frac{v_i}{v_j} = \frac{v^i}{v^j} = v^{i-j}$ only depends on $i - j$. In particular, using such $\mathbf{v}$, the number of preimages in the VC construction for linear functions will decrease from $O(w^2)$ to $O(w)$. More generally, adopting this approach would allow us to construct a VC scheme for degree-d polynomial maps with public parameters size $O(d \cdot w)$ instead of $O(w^{2d})$.

Starting from this key observation, we generalized it to overcome the drawbacks of both approaches to construct lattice-based SNARKs that we have mentioned before. In particular, among the result of this work are:

1. The first recursive folding (i.e., Bulletproofs-like) protocol for linear relations with *polylogarithmic* verifier runtime. Traditionally, the verifier runtime has been the efficiency bottleneck for such protocols (regardless of the underlying assumptions).
2. The first verifiable delay function (VDF) based on lattices, building on a recently introduced sequential relation.
3. The first lattice-based *linear-time prover* succinct argument for NP, in the preprocessing model. The soundness of the scheme is based on (knowledge)-k-R-ISIS assumption.

To achieve these results, we started by developing a new family of commitment schemes for committing to short vectors $\mathbf{x} \in \mathbb{Z}^d$ and a companion argument systems for proving that the committed vector is in fact a bit string, i.e., $\mathbf{x} \in \{0,1\}^d$, which is the main technical ingredient behind all of our results. In their simplest form, the commitment key is a single random element $v \leftarrow \mathbb{Z}_q^\times$. To commit to a *short* $\mathbf{x} \in \mathbb{Z}^d$, we interpret $\mathbf{x}$ as the coefficients of a degree-d polynomial $p_{\mathbf{x}}(V)$ without constant term, and compute the commitment as the evaluation of $p_{\mathbf{x}}$ at the point v modulo q, i.e.

$$p_{\mathbf{x}}(V) = \sum_{i=1}^{d} x_i \cdot V^i \qquad \text{and} \qquad c = p_{\mathbf{x}}(v) \bmod q.$$

We refer to this family of commitment schemes as vanishing short integer solution (vSIS) commitments. The binding property of the vSIS commitment above is based on the following vSIS assumption which we introduce in this work. Informally, the assumption says:

> Given a random point $v \leftarrow \mathbb{Z}_q^\times$, it is hard to find a degree-d polynomial $p = \sum_{i=0}^{d} p_i \cdot V^i \in \mathbb{Z}[V]$ with short coefficients such that $p(v) = 0 \bmod q$.

In other words, if we let $\mathbf{v} = (v, v^2, \ldots, v^d)^\intercal$, the assumption says that it is hard to find a short non-zero vector $\mathbf{p}$ such that $\langle \mathbf{p}, \mathbf{v} \rangle = 0 \bmod q$. That is, the assumption can be seen as a structured version of the SIS assumption: instead of having $\mathbf{v} \leftarrow \mathbb{Z}_q^d$, one has $\mathbf{v} = (v, v^2, \ldots, v^d)^\intercal$ for $v \leftarrow \mathbb{Z}_q^\times$.

In general, the vSIS assumption could be parametrised by a set $\mathcal{G}$ of (multivariate) monomials[10] over $\mathbb{Z}$, where the task is to find a short linear combination $(p_g)_{g \in \mathcal{G}}$ such that $\sum_{g \in \mathcal{G}} p_g \cdot g(\mathbf{v}) = 0 \bmod q$. In the work, we study the plausibility of this new assumption. In particular, we show that vSIS is no easier than the k-R-ISIS problem. We also show that vSIS can be explained as a natural generalisation of the search NTRU problem. We propose a worst-case to average-case reduction and a reduction from search NTRU, both conditioning on the hardness of decision NTRU.

The vSIS commitment schemes have nice properties.

- Succinct: The size of the commitment key and the commitment are logarithmic in the size of the input. In particular, this implies that the commitment is also a collision-resistant hash function with very short key.
- Homomorphic: The commitment is (bounded) linearly homomorphic and multiplicatively homomorphic for a constant number of multiplications.
- Foldable: We show that the commitment can be "folded" (in the sense of folding arguments, e.g., Bulletproofs [BLNS20]) in such a way that the folded commitment key retains a succinct representation.

Proof of Binary-Satisfiability of Linear Relations. An important relation in lattice-based cryptography is

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \qquad \text{and} \qquad \mathbf{x} \in \{0,1\}^d.$$

We now show how to construct a succinct argument system for a prover to convince a verifier that a vector $\mathbf{x} \in \mathbb{Z}^d$ satisfies the above equation. As building block, we will use succinct argument systems for SIS relations with soundness gap, i.e., they are complete and sound for relations of the form

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0,$$

[10] Or rational functions in general.

but the constraints on the shortness of $\mathbf{x}$ differ. That is, we will turn succinct arguments for showing that $\mathbf{x}$ satisfying a linear relation is short, into an argument for showing that $\mathbf{x}$ is *exactly binary*. While this may seem like a technicality, this *proof of binariness* will be crucial for our later applications, and can be generalised to prove arbitrary quadratic relations. Next, we will show how to instantiate the required building blocks using two different approaches.

The public parameters of our argument system contains a random vector $\mathbf{h} \in \mathbb{Z}_{q_1}^d$ and a vSIS commitment key $v \in \mathbb{Z}_{q_2}^\times$, where $q_0 \ll q_1 \ll q_2$ and the purpose of $\mathbf{h}$ will become clear later. For the sake of exposition, let

$$\mathbf{v} \coloneqq (v, v^2, \ldots, v^d)^\intercal, \qquad \text{and} \qquad \bar{\mathbf{v}} \coloneqq (v^{-1}, v^{-2}, \ldots, v^{-d})^\intercal,$$

For $\mathbf{x} \in \mathbb{Z}^d$ and some $\mathbf{w} = (\mathbf{w}_-, \mathbf{w}_+) \in \mathbb{Z}^{2d}$, consider the (Laurent) polynomials

$$\begin{aligned} p_{\mathbf{x}}(V) &= \sum_{i=1}^{d} x_i \cdot V^i, \\ p_{\mathbf{h}\circ\mathbf{x}}(V^{-1}) &= \sum_{i=1}^{d} h_i \cdot x_i \cdot V^{-i}, \quad \text{and} \\ p_{\mathbf{w}_-}(V^{-1}) + p_{\mathbf{w}_+}(V) &= \sum_{i=-d+1}^{-1} w_{-,i} \cdot V^i + \sum_{i=1}^{d} w_{+,i} \cdot V^i \end{aligned}$$

where $\mathbf{h} \circ \mathbf{x}$ denotes the Hadamard (component-wise) product of the two vectors. The argument proceeds as follows:

1. The prover reveals the following "complementary" vSIS commitments to $\mathbf{x}$:

$$\begin{aligned} c_{\mathbf{x}} &\coloneqq p_{\mathbf{x}}(v) = \langle \mathbf{x}, \mathbf{v} \rangle \bmod q_2, \quad \text{and} \\ \bar{c}_{\mathbf{x}} &\coloneqq p_{\mathbf{h}\circ\mathbf{x}}(v^{-1}) = \langle \mathbf{h} \circ \mathbf{x}, \bar{\mathbf{v}} \rangle \bmod q_2. \end{aligned}$$

2. The prover then proves the following relations:

$$\exists\, \mathbf{x} \in \mathbb{Z}^d, \quad \begin{aligned} \mathbf{M} \cdot \mathbf{x} &= \mathbf{y} \bmod q_0, \\ p_{\mathbf{x}}(v) &= c_{\mathbf{x}} \bmod q_2, \quad \text{and} \quad \|\mathbf{x}\| \approx 0. \\ p_{\mathbf{h}\circ\mathbf{x}}(v^{-1}) &= \bar{c}_{\mathbf{x}} \bmod q_2, \end{aligned} \tag{1.2}$$

$$\exists\, \mathbf{w} \in \mathbb{Z}^{2d}, \quad p_{\mathbf{w}_-}(v^{-1}) + p_{\mathbf{w}_+}(v) = c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - p_{\mathbf{h}\circ\mathbf{1}}(v^{-1})) \bmod q_2 \quad \text{and} \quad \|\mathbf{w}\| \approx 0. \tag{1.3}$$

Since $p_{\mathbf{x}}(v)$, $p_{\mathbf{h}\circ\mathbf{x}}(v)$, and $p_{\mathbf{w}_-}(v^{-1}) + p_{\mathbf{w}_+}(v)$ can be computed as linear functions evaluated at the monomial expansion of v, eqs. (1.2) and (1.3) can be proven by using argument systems for SIS relations, as required above.

The interesting bit of our protocols is that, even though the underlying arguments for the SIS relation have soundness gaps, the verifier of our protocol will be convinced that $\mathbf{x}$ is *exactly* binary.

First, from the knowledge soundness of the argument for eq. (1.2), and eq. (1.3), the verifier is convinced that there exists candidate short vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{w}}$ satisfying eq. (1.2) and eq. (1.3) respectively. In particular, from eq. (1.2), one has that the extracted witness $\hat{\mathbf{x}}$ satisfies

$$p_{\hat{\mathbf{x}}}(v) = \sum_{i=1}^{d} \hat{x}_i \cdot v^i = c_{\mathbf{x}} \bmod q_2, \quad \text{and}$$
$$p_{\mathbf{h} \circ \hat{\mathbf{x}}}(v^{-1}) = \sum_{i=1}^{d} h_i \cdot \hat{x}_i \cdot v^{-1} = \bar{c}_{\mathbf{x}} \bmod q_2.$$

It follows that

$$\begin{aligned} c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - p_{\mathbf{h} \circ \mathbf{1}}(v)) &= \left(\sum_{i=1}^{d} \hat{x}_i \cdot v^i \right) \cdot \left(\sum_{j=1}^{d} h_j \cdot \hat{x}_j \cdot v^{-j} - \sum_{j=1}^{d} h_j \cdot v^{-j} \right) \\ &= \left(\sum_{i=1}^{d} \hat{x}_i \cdot v^i \right) \cdot \left(\sum_{j=1}^{d} h_j \cdot (\hat{x}_j - 1) \cdot v^{-j} \right) \\ &= \sum_{i,j \in [d]} h_j \cdot \hat{x}_i \cdot (\hat{x}_j - 1) \cdot v^{i-j} \\ &= \sum_{h=-d+1}^{d-1} \underbrace{\left(\sum_{\substack{i,j \in [d] \\ i-j=h}} h_j \cdot \hat{x}_i \cdot (\hat{x}_j - 1) \right)}_{=:\hat{u}_h} \cdot v^h. \end{aligned}$$

In particular, from $\hat{\mathbf{x}}$, one could derive a somewhat short vector $\hat{\mathbf{u}} = (\hat{\mathbf{u}}_-, \hat{u}_0, \hat{\mathbf{u}}_+) \in \mathbb{Z}^{2d-1}$ such that

$$\sum_{h=-d+1}^{d-1} \hat{u}_h \cdot v^h = c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - p_{\mathbf{h} \circ \mathbf{1}}(v^{-1})) \bmod q_2.$$

At the same time, from eq. (1.3), we know that $\hat{\mathbf{w}} = (\hat{\mathbf{w}}_-, \hat{\mathbf{w}}_+) \in \mathbb{Z}^{2d-2}$ is a short vector such that

$$\sum_{i=-d+1}^{-1} \hat{w}_{-,i} \cdot v^i + \sum_{i=1}^{d-1} \hat{w}_{+,i} \cdot v^i = c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - p_{\mathbf{h} \circ \mathbf{1}}(v^{-1})) \bmod q_2.$$

This means that

$$\begin{aligned}\sum_{h=-d+1}^{d-1} \hat{u}_h \cdot v^h - \sum_{i=-d+1}^{-1} \hat{w}_{-,i} \cdot v^i - \sum_{i=1}^{d-1} \hat{w}_{+,i} \cdot v^i \\ = \sum_{i=-d+1}^{-1} (\hat{u}_i - \hat{w}_{-,i}) \cdot v^i + \hat{u}_0 \cdot v^0 + \sum_{i=1}^{d-1} (\hat{u}_i - \hat{w}_{+,i}) \cdot v^i \\ = 0 \bmod q_2.\end{aligned}$$

In particular, the vector $(\hat{\mathbf{u}}_- - \hat{\mathbf{w}}_-, \hat{u}_0, \hat{\mathbf{u}}_+ - \hat{\mathbf{w}}_+)$ defines a Laurent polynomial with short coefficients which vanishes at v. If such a vector is non-zero, then it yields a non-zero short solution to a vSIS problem (in this case it corresponds to the SIS problem for the structured vector $(v^{-d+1}, \ldots, v^{-1}, v^0, v, \ldots, v^{d-1})$), which we assume to be hard. We deduce that such vector must be zero. In particular, the middle term $\hat{u}_0$ is equal to zero. By definition of $\hat{u}_h$, we have

$$\hat{u}_0 = \langle \hat{\mathbf{x}}, \mathbf{h} \circ (\hat{\mathbf{x}} - \mathbf{1}) \rangle = \sum_{i=1}^{d} h_i \cdot \underbrace{\hat{x}_i \cdot (\hat{x}_i - 1)}_{=0 \text{ iff } \hat{x}_i \in \{0,1\}}.$$

The fact that $\hat{u}_0 = 0$ *does not* directly imply that all of its summands are also zero (which is what we need to ensure that $\hat{\mathbf{x}}$ is binary). This is where the vector $\mathbf{h}$ comes into play, using a technique first introduced in [ACL+22]: Since $\hat{u}_0 = 0$, then we also have $\hat{u}_0 = \sum_{i=1}^{d} h_i \cdot \hat{x}_i \cdot (\hat{x}_i - 1) = 0 \bmod q_1$. If $\hat{\mathbf{x}}$ is not binary, the vector $\hat{\mathbf{x}} \circ (\hat{\mathbf{x}} - \mathbf{1})$ would be a short non-zero solution to the SIS instance given by $\mathbf{h}$ over $\mathbb{Z}_{q_1}$. Therefore, $\hat{x}_i \cdot (\hat{x}_i - 1) = 0$ for all $i \in [d]$. Thus, the extracted witness $\hat{\mathbf{x}}$ must be binary as required.

1.1.4 Efficient Proofs for SIS Relations

In the above proof of binary-satisfiability of linear relations, the prover and verifier computation costs are dominated by the costs of the succinct arguments for SIS relations with soundness gaps. Here we discuss how we can use the two approaches in the literature, and improve on both fronts using the algebraic properties of our vSIS-based commitment scheme.

Approach I: Polylogarithmic Verifier for Structured Relations. While we cannot hope to reduce the verifier complexity for general matrices $\mathbf{M}$, for suitably structured $\mathbf{M}$ the verification can be sped up to run in time polylogarithmic in the witness length. As an example, the simplest $\mathbf{M}$ with the required structure is a vector consisting of powers of an element $v \in \mathbb{Z}_q^\times$, i.e.,

$$\mathbf{M} = \begin{pmatrix} v & v^2 & \ldots & v^d \end{pmatrix} \bmod q.$$

To see why this is the case, it suffices to observe that the verifier complexity is dominated by the computation of the matrix $\mathbf{M}^{(\ell)}$, defined recursively in eq. (1.1), and obtained by

successive foldings of the starting matrix $\mathbf{M}^{(0)} = \mathbf{M}$. Plugging in the structured relation, we can see that at each iteration the matrix evolves into

$$\begin{aligned}\mathbf{M}^{(i+1)} = \mathbf{M}_L^{(i)} + \mathbf{M}_R^{(i)} \cdot r_i^{-1} &= \begin{pmatrix} v & v^2 & \dots & v^{d_i/2} \end{pmatrix} + \begin{pmatrix} v^{d_i/2+1} & v^{d_i/2+2} & \dots & v^{d_i} \end{pmatrix} \cdot r_i^{-1} \\ &= \begin{pmatrix} v & v^2 & \dots & v^{d_i/2} \end{pmatrix} + \begin{pmatrix} v & v^2 & \dots & v^{d_i/2} \end{pmatrix} \cdot v^{d_i/2} \cdot r_i^{-1} \\ &= \begin{pmatrix} v & v^2 & \dots & v^{d_i/2} \end{pmatrix} \cdot (1 + v^{d_i/2} \cdot r_i^{-1}) \bmod q,\end{aligned}$$

where d_i is the input length at the i-th iteration. Recursing over all iterations, we obtain that the final matrix $\mathbf{M}^{(\ell)}$ is defined as

$$\mathbf{M}^{(\ell)} = \prod_{i=0}^{\ell-1} \left(1 + v^{2^{\ell-i-1}} \cdot r_i^{-1}\right) \bmod q,$$

which can be computed in time polynomial in ℓ, i.e., polylogarithmic in d. In our work we extend the above structured folding technique in three ways:

1. We identify a general class of "foldable" (block-)matrices for which the verifier computation can be made polylogarithmic in the number of columns.
2. By modifying the Bulletproofs protocol with techniques borrowed from another folding protocol of Pietrzak [Pie19], we are able to support foldable matrices with an arbitrary (i.e., non-power-of-2) number of columns, without breaking the foldable structure.[11]
3. Borrowing techniques from [Pie19] again, we can make the verifier computation *also* polylogarithmic in the number of rows of $\mathbf{M}$, for $\mathbf{M}$ with repeating block-bidiagonals, if $\mathbf{y}$ is also foldable.

Approach II: Achieving Quasi-Linear Time Prover. The second approach for lattice-based arguments for SIS relations is the recent work of [ACL+22], which is based on the newly introduced (knowledge-)k-R-ISIS assumption. As we have already mentioned, a major drawback of this approach is that the public parameters size and the prover complexity are at least quadratic in the relation size. A natural idea is to choose $\mathcal{G} = \{ V_1, V_2, \dots, V_d \}$ with $\mathbf{v} = (v, v^2, \dots, v^d)$.[12] This makes

$$|\{ (g \cdot (g')^{-1})(\mathbf{v}) : g, g' \in \mathcal{G}, g \neq g' \}| = |\{ v^{-i}, v^i \}_{i=1}^{d-1}| = 2d - 2 = O(d).$$

Further exploiting fast multiplication algorithms for Toeplitz matrices allows us to achieve quasi-linear prover time. In Chapter 4, we also show how to support natively modular arithmetic, by borrowing techniques from chainable functional commitments [GR19, BCFL22].

[11] The usual technique of padding zero columns breaks the foldable structure.
[12] The same result could be obtained setting $\mathcal{G} = \{ V^1, V^2, \dots, V^d \}$ and $\mathbf{v} = v$.

Putting everything together, we obtain SNARKs for quadratic relations with quasi-linear-time prover and polylogarithmic-time verifier (after preprocessing for the unstructured case). We highlight two particular instances.

First, we obtain SNARKs for proving "$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\mathbf{x}$ is *exactly binary*". In particular, applying the structured instantiation on the recently introduced SIS-based sequential relations [LM23], we obtain the first lattice-based verifiable delay functions (VDF). Prior lattice-based schemes [YAZ+19, BLS19, ENS20, LNP22] for exact SIS relations[13] are not succinct.

Second, we obtain SNARKs for rank-1 constraint satisfiability (R1CS). Prior lattice-based schemes [ACL+22, BCFL22] have at least quadratic-time provers.

1.2 On Lattice-Based Knowledge Assumptions

In both [ACL+22] and [CLM23] part of the results are based on the newly introduced knowledge k-R-ISIS assumption [ACL+22]. A recent result of Wee and Wu [WW23a] shows a counterexample that morally/plausibly invalidates this knowledge assumption.

Before describing the result of Wee and Wu [WW23a], let us briefly recall definition of falsifiable and non-falsifiable assumptions.

For almost any interesting cryptographic task we need to make some computational hardness assumptions. Such computational hardness assumptions can be partitioned into two classes: falsifiable assumptions and non-falsifiable ones. Informally, we say that an assumption is falsifiable if it can be modeled as an interactive game between an adversary and an efficient challenger that can efficiently decide if the adversary won the game [GW11]. Such definition captures the fact that we can efficiently check if an adversarial strategy breaks the assumption. On the contrary, non-falsifiable assumptions are assumptions that do not lend themselves easily to "efficient falsification" [Nao03].

However, even non-falsifiable assumption can in fact be (conditionally) falsified. For example Bellare and Palacio [BP04] showed that, assuming the hardness of the discrete logarithm problem, the knowledge of exponent assumption introduced by Hada and Tanaka [HT98] is false.

In the lattice setting, before [ACL+22], the only other knowledge assumption was that (adaptations of) Regev's encryption scheme [Reg05] are linear-only [BCI+13, BISW17], i.e, that given a public key pk and ciphertexts $(c_1, \ldots, c_m)$ of any such encryption scheme, it is infeasible to compute a new ciphertext c' in the image of $\mathsf{Enc}(\mathsf{pk}, \cdot)$, except by evaluating an affine combination of the ciphertexts $(c_1, \ldots, c_m)$. This assumption was used to construct designated-verifier lattice-based SNARKs [GMNO18, ISW21]. Notice that this knowledge assumption is not "efficiently falsifiable": to falsify it, one should exhibit an adversary for which there does not exists an extractor; but the non-existence of such an algorithm cannot be checked efficiently.

[13] not counting those for more general relations

Let us now discuss the recent findings of Wee and Wu [WW23a] regarding the cryptanalysis of the knowledge-k-R-ISIS assumption. At the end of this section, we will explore the implications of their results on the contributions that have presented previously.

We start by recalling what the knowledge k-R-ISIS assumptions says. Somewhat informally, the assumption says that for any PPT algorithm $\mathcal{A}$ that on input

$$\mathbf{B} \in \mathbb{Z}_q^{n\times m}, \mathbf{T} \in \mathbb{Z}_q^{n\times t}, \{\mathbf{v}_i\}_{i\in[k]} \in (\mathbb{Z}_q^t)^k, \{\mathbf{u}_i\}_{i\in[k]} \in (\mathbb{Z}^m)^k,$$ [14]

where $\mathbf{u}_i \in \mathbf{B}^{-1}(\mathbf{T}\cdot\mathbf{v}_i)$ and $\|\mathbf{u}_i\| \approx 0$[15], i.e., for each $i \in [k]$ one has

$$\mathbf{B}\cdot\mathbf{u}_i = \mathbf{T}\cdot\mathbf{v}_i \bmod q \qquad \text{and} \qquad \|\mathbf{u}_i\| \approx 0,$$

outputs $(\mathbf{u},\mathbf{v}) \in \mathbb{Z}^m \times \mathbb{Z}_q^t$ such that

$$\mathbf{B}\cdot\mathbf{u} = \mathbf{T}\cdot\mathbf{v} \mod q \qquad \text{and} \qquad \|\mathbf{u}\| \approx 0,$$

then, there exists a PPT extractor $\mathcal{E}_\mathcal{A}$ that on input $\mathcal{A}$'s input, output $\mathbf{x} \in \mathbb{Z}^k$ such that

$$\mathbf{v} = \sum_i \mathbf{v}_i \cdot x_i \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0.$$

The underlying idea was that the only way a PPT algorithm could produce a preimage under $\mathbf{B}$ of a vector in the column span of $\mathbf{T}$ was by taking a short linear combination[16] of the given preimage $\mathbf{u}_i$'s.

The matrix $\mathbf{T}$ is used to "sparsify" the allowed image output. Indeed, the same assumption without the matrix $\mathbf{T}$ is clearly false, as the adversary $\mathcal{A}$ could sample a random short vector $\mathbf{u}$ and output $(\mathbf{u}, \mathbf{B}\cdot\mathbf{u})$. And all this without using the given hints. For the same reason, we need $\mathbf{T}$ to be a tall matrix, i.e., $n > t$, so that the column space of $\mathbf{T}$ is a proper subspace of $\mathbb{Z}_q^n$, and the image, $\mathbf{B}\cdot\mathbf{u}$, of a random short vector $\mathbf{u}$ is with overwhelming probability not in the column span of $\mathbf{T}$.

To show that knowledge k-R-ISIS is false, one should show there is an adversary for which there exists no extractor. Wee and Wu identify an adversary for which there does not appear to exist an extractor.

The key observation in their analysis is the following: the equation

$$\mathbf{B}\cdot\mathbf{u}_i = \mathbf{T}\cdot\mathbf{v}_i \bmod q,$$

can be rewritten as

$$\begin{bmatrix}\mathbf{B} & \mathbf{T}\end{bmatrix} \cdot \begin{bmatrix}\mathbf{u}_i \\ -\mathbf{v}_i\end{bmatrix} = \mathbf{0} \mod q.$$

[14] In the previous formulation of the assumption, we had $t = 1$.

[15] To avoid introducing too many parameters already in the introduction, by $\|\mathbf{u}\| \approx 0$ we will mean that $\mathbf{u}$ is short.

[16] linear combinations where the coefficients used have small norm

Using the so-called gadget matrix $\mathbf{G} =: \mathbf{g} \otimes \mathbf{I}_n$, where $\mathbf{g} = [1, 2, \ldots, 2^{\lceil \log q \rceil - 1}]$, and $\mathbf{G}^{-1}(\cdot)$ indicating the binary decomposition operator, the above equation can be rewritten as

$$\begin{bmatrix} \mathbf{B} & \mathbf{T} \cdot \mathbf{G} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u}_i \\ -\mathbf{G}^{-1}(\mathbf{v}_i) \end{bmatrix} = \mathbf{0} \mod q.$$

Since $\mathbf{u}_i$ is short by assumption, and $\mathbf{G}^{-1}(\mathbf{v}_i)$ is short, as it is the binary decomposition of $\mathbf{v}_i$, we have that

$$\left\| \begin{bmatrix} \mathbf{u}_i \\ -\mathbf{G}^{-1}(\mathbf{v}_i) \end{bmatrix} \right\| \approx 0,$$

for each $i \in [k]$. Since the $(\mathbf{u}_i, \mathbf{v}_i)$'s are independently generated, if $k \geq m + t\lceil \log q \rceil$, one could *heuristically assume* that the vectors

$$\underbrace{\begin{bmatrix} \mathbf{u}_1 & \ldots & \mathbf{u}_{m+t\lceil \log q \rceil} \\ -\mathbf{G}^{-1}(\mathbf{v}_1) & \ldots & -\mathbf{G}^{-1}(\mathbf{v}_{m+t\lceil \log q \rceil}) \end{bmatrix}}_{=:\mathbf{Z}} \in \mathbb{Z}^{(m+t\lceil \log q \rceil) \times (m+t\lceil \log q \rceil)}$$

are linearly independent over the reals, i.e., $\mathbf{Z}^{-1} \in \mathbb{R}^{(m+t\lceil \log q \rceil) \times (m+t\lceil \log q \rceil)}$ exists. If that is the case, then $\mathbf{Z}$ is an Ajtai trapdoor [Ajt96] for the matrix $\begin{bmatrix} \mathbf{B} & \mathbf{T} \cdot \mathbf{G} \end{bmatrix}$. Given any such trapdoor, it is possible to sample short vectors $\begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix}$, such that $\begin{bmatrix} \mathbf{B} & \mathbf{T} \cdot \mathbf{G} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix} = \mathbf{0} \mod q$. For example, one possible way is by using Babai's rounding algorithm [Bab86]:

- by Gaussian elimination, obtain an arbitrary (non-zero) vector $\mathbf{x} \in \mathbb{Z}^{m+t\lceil \log q \rceil}$ such that $\begin{bmatrix} \mathbf{B} & \mathbf{T} \cdot \mathbf{G} \end{bmatrix} \cdot \mathbf{x} = \mathbf{0} \mod q$,
- output $\mathbf{x} - \mathbf{Z} \cdot \lfloor \mathbf{Z}^{-1} \cdot \mathbf{x} \rfloor \in \mathbb{Z}^{m+t\lceil \log q \rceil}$.

By construction, we have

$$\begin{aligned} \begin{bmatrix} \mathbf{B} & \mathbf{T} \cdot \mathbf{G} \end{bmatrix} \cdot (\mathbf{x} - \mathbf{Z} \cdot \lfloor \mathbf{Z}^{-1} \cdot \mathbf{x} \rfloor) &= \underbrace{\begin{bmatrix} \mathbf{B} & \mathbf{T} \cdot \mathbf{G} \end{bmatrix} \cdot \mathbf{x}}_{=\mathbf{0} \mod q} - \underbrace{\begin{bmatrix} \mathbf{B} & \mathbf{T} \cdot \mathbf{G} \end{bmatrix} \cdot \mathbf{Z}}_{=\mathbf{0} \mod q} \cdot \lfloor \mathbf{Z}^{-1} \cdot \mathbf{x} \rfloor \\ &= \mathbf{0} \mod q. \end{aligned}$$

Moreover

$$\begin{aligned} \left\| \mathbf{x} - \mathbf{Z} \cdot \lfloor \mathbf{Z}^{-1} \cdot \mathbf{x} \rfloor \right\| &= \left\| \mathbf{Z} \cdot \mathbf{Z}^{-1} \cdot \mathbf{x} - \mathbf{Z} \cdot \lfloor \mathbf{Z}^{-1} \cdot \mathbf{x} \rfloor \right\| \\ &= \left\| \mathbf{Z} \cdot (\mathbf{Z}^{-1} \cdot \mathbf{x} - \lfloor \mathbf{Z}^{-1} \cdot \mathbf{x} \rfloor) \right\| \\ &= \left\| \mathbf{Z} \cdot \{\mathbf{Z}^{-1} \cdot \mathbf{x}\} \right\| \\ &\approx 0, \end{aligned}$$

where $\{\cdot\}$ denotes the fractional part operator and in the last line we have used that both $\mathbf{Z}$ and $\{\mathbf{Z}^{-1} \cdot \mathbf{x}\}$ have small norm. Using this procedure, an algorithm can produce a tuple $(\mathbf{u}, \mathbf{v})$ that satisfies the required constrains without apparently performing short (integer) linear combination of the given preimages. However, one should also note that in all application of this knowledge assumption in [ACL+22] and [CLM23], one has $k \gg m + t\lceil \log q \rceil$. This means that more short vectors in the kernel of $\begin{bmatrix}\mathbf{B} & \mathbf{T} \cdot \mathbf{G}\end{bmatrix}$ are provided, than those required to construct an Ajtai trapdoor. If this is the case, then the extractor has more flexibility in coming up with a short linear combination that would explain the adversary's output, as more short linear combinations are possible. In other words, for this regime of parameters, the existence of an extractor is more conceivable.

Implications on [ACL+22] and [CLM23]. As already mentioned before, the result of Wee and Wu [WW23a] plausibly invalidates the knowledge k-R-ISIS assumption. In particular, we could not come up with an extractor for such an adversary, and it is even possible that such an extractor does not in fact exist.

At the same time, even though [WW23a] show that this translates to a direct attack against (variant of) the linear functional commitment scheme, [WW23a] also argue that this does not seem to extend to higher-degree polynomials or to the SNARK construction itself. As far as we are aware, these later constructions could still be considered "heuristically" extractable, even though this property cannot not be reduced to a clear assumption anymore.

In fact, with regards to the construction of a lattice-based SNARK for NP, assuming that the VC construction for degree-2 polynomial maps from [ACL+22] is extractable would suffice. Since this assumption is independent of the specific language for which the SNARK would then be used, it could be seen as a plausible knowledge assumption from which one can construct a publicly-verifiable lattice-based SNARK.

Nevertheless, this assumption is not completely satisfying. We hope that our attempt to define a (publicly verifiable) knowledge assumption in the lattice world, and the follow-up negative result of Wee and Wu, will urge more research effort into coming up with the right formulation for this kind of assumptions.

CHAPTER 2

CCA-Secure (Puncturable) KEMs from Encryption with Non-Negligible Decryption Errors

Abstract

Public-key encryption (PKE) schemes or key-encapsulation mechanisms (KEMs) are fundamental cryptographic building blocks to realize secure communication protocols. There are several known transformations that generically turn weakly secure schemes into strongly (i.e., IND-CCA) secure ones. While most of these transformations require the weakly secure scheme to provide perfect correctness, Hofheinz, Hövelmanns, and Kiltz (HHK) (TCC 2017) have recently shown that variants of the Fujisaki-Okamoto (FO) transform can work with schemes that have negligible correctness error in the (quantum) random oracle model (QROM). Many recent schemes in the NIST post-quantum competition (PQC) use variants of these transformations. Some of their CPA-secure versions even have a non-negligible correctness error and so the techniques of HHK cannot be applied.

In this work, we study the setting of generically transforming PKE schemes with potentially large, i.e., non-negligible, correctness error to ones having negligible correctness error. While there have been previous treatments in an asymptotic setting by Dwork, Naor, and Reingold (EUROCRYPT 2004), our goal is to come up with practically efficient compilers in a concrete setting and apply them in two different contexts: firstly, we show how to generically transform weakly secure deterministic or randomized PKEs into CCA-secure KEMs in the (Q)ROM using variants of HHK. This applies to essentially all candidates to the NIST PQC based on lattices and codes with non-negligible error, for which we provide an extensive analysis. We thereby show that it improves some of the

code-based candidates. Secondly, we study puncturable KEMs in terms of the Bloom Filter KEM (BFKEM) proposed by Derler et al. (EUROCRYPT 2018) which inherently have a non-negligible correctness error. BFKEMs are a building block to construct fully forward-secret zero round-trip time (0-RTT) key-exchange protocols. In particular, we show the first approach towards post-quantum secure BFKEMs generically from lattices and codes by applying our techniques to identity-based encryption (IBE) schemes with (non-)negligible correctness error.

This chapter presents the first result of the collaboration with Sebastian Ramacher, Daniel Slamanig, and Christoph Striecks and was published at the 26th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'20) under the title "CCA-Secure (Puncturable) KEMs from Encryption with Non-Negligible Decryption Errors" [CRSS20]. I am mainly responsible for the analysis of the different compilers and the linked security proofs. I am also responsible for writing the corresponding sections of the chapter. The accompanying appendix contains omitted proofs and efficiency evaluations.

2.1 Introduction

Public-key encryption (PKE) schemes or key-encapsulation mechanisms (KEMs) are fundamental cryptographic building blocks to realize secure communication protocols. The security property considered standard nowadays is security against chosen-ciphertext attacks (IND-CCA security). This is important to avoid pitfalls and attacks in the practical deployments of such schemes, e.g., padding-oracle attacks as demonstrated by Bleichenbacher [Ble98] and still showing up very frequently [JSS12, ASS+16, BSY18, RGG+19]. Also, for key exchange protocols that achieve the desirable forward-secrecy property, formal analysis shows that security against active attacks is required (cf. [JKSS12, KPW13, DFGS15, PST20]). This equally holds for recent proposals for fully forward-secret zero round-trip time (0-RTT) key-exchange protocols from puncturable KEMs [GHJL17, DJSS18, DGJ+21] and even for ephemeral KEM keys for a post-quantum secure TLS handshake without signatures [SSW20a].

In the literature, various different ways of obtaining CCA security generically from weaker encryption schemes providing only chosen-plaintext (IND-CPA) or one-way (OW-CPA) security are known. These can be in the standard model using the double-encryption paradigm due to Naor and Yung [NY90], the compiler from selectively secure identity-based encryption (IBE) due to Canetti, Halevi and Katz [CHK04], or the more recent works due to Koppula and Waters [KW19] based on so called hinting pseudo-random generators and Hohenberger, Koppula, and Waters [HKW20] from injective trapdoor functions. In the random oracle model (ROM), CCA security can be generically obtained via the well-known and widely-used Fujisaki-Okamoto (FO) transform [FO99, FO13] yielding particularly practical efficiency.

Perfect correctness and (non-)negligible correctness error. A property common to many compilers is the requirement for the underlying encryption schemes to provide

perfect correctness, i.e., there are no valid ciphertexts where the decryption algorithm fails when used with honestly generated keys. Recently, Hofheinz, Hövelmanns, and Kiltz (HHK) [HHK17a] investigated different variants of the FO transform also in a setting where the underlying encryption scheme has non-perfect correctness and in particular decryption errors may occur with a negligible probability in the security parameter. This is interesting since many PKE schemes or KEMs based on conjectured quantum-safe assumptions and in particular assumptions on lattices and codes do not provide perfect correctness. Even worse, some of the candidates submitted to the NIST post-quantum competition (PQC) suffer from a *non-negligible* correctness error and so the FO transforms of HHK cannot be applied. Ad-hoc approaches to overcome this problem that are usually chosen by existing constructions in practice — if the problem is considered at all — is to increase the parameters to obtain a suitably small decryption error, applying an error correcting code on top or implementing more complex decoders. In practice, these ad-hoc methods come with drawbacks. Notably, LAC, which is a Learning With Errors (LWE) based IND-CCA secure KEM in the 2nd round of the NIST PQC that applies an error correcting code, is susceptible to a key-recovery attack recently proposed by Guo et al. [GJY19]. Also, code-based schemes have a history of attacks [GJS16, SSPB19, FHS+17] due to decoding errors. Recently, Bindel and Schanck [BS20] proposed a failure boosting attack for lattice-based schemes with a non-zero correctness error. For some code-based schemes, the analysis of the decoding error is a non-trivial task as it specifically depends on the decoder. For instance, the analysis of BIKE's decoder, another 2nd round NIST PQC candidate, has recently been updated [SV19].

*C*onsequently, it would be interesting to have rigorous and simple approaches to remove decryption errors (to a certain degree) from PKE schemes and KEMs.

Immunizing encryption schemes. The study of "immunizing" encryption schemes from decryption errors is not new. Goldreich, Goldwasser, and Halevi [GGH97] studied the reduction or removal of decryption errors in the Ajtai-Dwork encryption scheme as well as Howgrave-Graham et al. [HNP+03] in context of NTRU. The first comprehensive and formal treatment has been given by Dwork, Naor, and Reingold [DNR04] who study different amplification techniques in the standard and random oracle model to achieve non-malleable (IND-CCA secure) schemes. One very intuitive compiler is the direct product compiler $\mathsf{Enc}^{\otimes\ell}$ which encrypts a message M under a PKE $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ with a certain decryption error δ under ℓ independent public keys from KGen, i.e,. $\mathsf{pk}' := (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$ as $\mathsf{Enc}'(\mathsf{pk}', M) := (\mathsf{Enc}(\mathsf{pk}_1, M), \ldots, \mathsf{Enc}(\mathsf{pk}_\ell, M))$. Dec', given $C' = (C_1, \ldots, C_\ell)$ tries to decrypt C_i, $1 \leq i \leq \ell$, and returns the result of a majority vote among all decrypted messages, yielding an encryption scheme with some error $\delta' \leq \delta$. Their asymptotic analysis, however, and limitation to PKEs with a binary message space does not make it immediate what this would mean in a concrete setting and in particular how to choose ℓ for practically interesting values of δ and δ'. For turning a so-obtained amplified scheme with negligible correctness error into a CCA-secure one in the ROM, they provide a transform using similar ideas, but more involved than the FO transform.

Bitansky and Vaikuntanathan [BV17] go a step further and turn encryption schemes with a correctness error into perfectly correct ones, whereas they even consider getting completely rid of bad keys (if they exist) and, thus, completely immunize encryption schemes. They build upon the direct product compiler of Dwork et al. and then apply reverse randomization [Nao90] and Nisan-Wigderson style derandomization [NW94]. Thereby, they partition the randomness space into good and bad randomness, and ensure that only good randomness is used for encryption and key generation.

Our goals. In this work, we are specifically interested in transformations that lift weaker schemes with non-negligible correctness error into CCA-secure ones with negligible error. Thereby, our focus is on modular ways of achieving this and can be seen as a concrete treatment of ideas that have also be discussed by Dwork et al. [DNR04], who, however, treat their approaches in an asymptotic setting only. We show that the direct product compiler can be used with variants of the standard FO transform considered by HHK [HHK17a] (in the ROM) as well as Bindel et al. [BHH+19] and Jiang et al. [JZM19] (in the quantum ROM (QROM) [BDF+11]). They are used by many candidates of the NIST PQC, when starting from PKE schemes having non-negligible correctness error generically. As we are particularly interested in *practical compilers* in a *concrete setting* to obtain CCA security for KEMs in the (Q)ROM, we analyze the concrete overhead of this compiler and its use with widely used variants of the transforms from HHK. Moreover, we provide a rigorous treatment of non-black-box applications of these ideas and show that they yield better concrete results than the direct application of the direct product compiler. Importantly, it gives a generic way to deal with the error from weaker schemes (e.g., IND-CPA secure ones with non-negligible error) which are easier to design. An interesting question that we will study is how does increasing from one to ℓ ciphertexts compare to increasing the parameters at comparable resulting decryption errors for existing round-two submissions in the NIST PQC. As it turns out, our approach performs well in context of code-based schemes but gives less advantage for lattice-based schemes.

We also study our approach beyond conventional PKE schemes and KEMs. In particular, a class of KEMs that have recently found interest especially in context of full forward-secrecy for zero round-trip time (0-RTT) key-exchange (KE) protocols are so-called *puncturable KEMs* [GM15, GHJL17, DJSS18, SSS+20] and, in particular, Bloom Filter KEMs (BFKEMs) [DJSS18, DGJ+21]. BFKEMs schemes are CCA-secure KEMs that inherently have non-negligible correctness error. Interestingly, however, the non-negligible correctness error comes from the Bloom filter layer and the underlying IBE scheme (specifically, the Boneh-Franklin [BF01] instantiation in [DJSS18]) is required to provide perfect correctness. Thus, as all post-quantum IBEs have at least negligible correctness error, there are no known post-quantum BFKEMs.

2.1.1 Contribution

Our contributions on a more technical level can be summarized as follows:

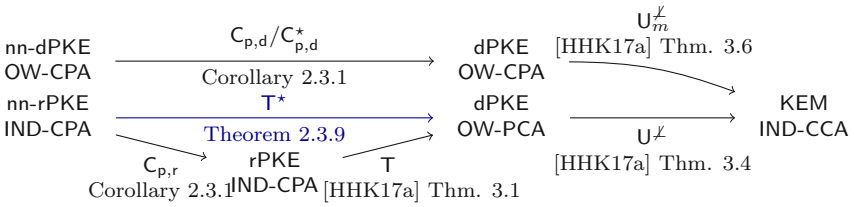

Figure 2.1: Overview of the transformations in the ROM with the results related to $\mathsf{T}^{\star}$ highlighted in blue. rPKE denotes a randomized PKE. dPKE denotes a deterministic PKE. The prefix nn indicates encryption schemes with non-negligible correctness error.

Generic transform. We revisit the ideas of the direct product compiler of Dwork et al. [DNR04] (dubbed $\mathsf{C}_{\mathsf{p,r}}$ and $\mathsf{C}_{\mathsf{p,d}}$ for randomized and deterministic PKEs, respectively) in the context of the modular framework of HHK [HHK17a]. In particular, we present a generic transform dubbed $\mathsf{T}^{\star}$ that, given any randomized PKE scheme with non-negligible correctness error, produces a derandomized PKE scheme with negligible correctness error. We analyze the transform both in the ROM and QROM and give a tight reduction in the ROM and compare it to a generic application of the direct product compiler. The transform naturally fits into the modular framework of HHK [HHK17a], and, thus, by applying the $\mathsf{U}^{\not\perp}$ transform, gives rise to an IND-CCA-secure KEM. For the analysis in the QROM, we follow the work of Bindel et al. [BHH+19]. We show that the $\mathsf{T}^{\star}$ transform also fits into their framework. Hence, given the additional injectivity assumption, we also obtain a tight proof for $\mathsf{U}^{\not\perp}$. But even if this assumption does not hold, the non-tight proofs of Jiang et al. [JZM19] and Hövelmanns et al. [HKSU20] still apply. Compared to the analysis of the T transform that is used in the modular frameworks, our reductions lose a factor of ℓ, i.e., the number of parallel ciphertexts required to reach a negligible correctness error, in the ROM and a factor of ℓ^2 in the QROM. For concrete schemes, this number is small (e.g., ≤ 5) and, thus, does not impose a significant loss. An overview of the transformations and how our transform fits into the modular frameworks is given in Figure 2.1 (ROM) and Figure 2.2 (QROM). Furthermore, using ideas similar to $\mathsf{T}^{\star}$, we discuss a modified version of the deterministic direct product compiler $\mathsf{C}_{\mathsf{p,d}}$ which we denote by $\mathsf{C}^{\star}_{\mathsf{p,d}}$, that compared to the original one allows to reduce the number of parallel repetitions needed to achieve negligible correctness error.

Evaluation. We evaluate $\mathsf{T}^{\star}$ based on its application to code- and lattice-based second-round candidates in the NIST PQC. In particular, we focus on schemes that offer IND-CPA secure versions with non-negligible correctness error such as ROLLO [ABD+19], BIKE [ABB+19], and Round5 [GZB+19]. We compare their IND-CCA variants with our transform applied to the IND-CPA schemes. In particular, for the code-based schemes such as ROLLO we can observe improvements in the combined size of public keys and ciphertexts, a metric important when used in protocols such as TLS, as well as its runtime efficiency. We also argue the ease of implementing our so-obtained schemes which can rely on simpler decoders. For lattice-based constructions, we find that the use of the

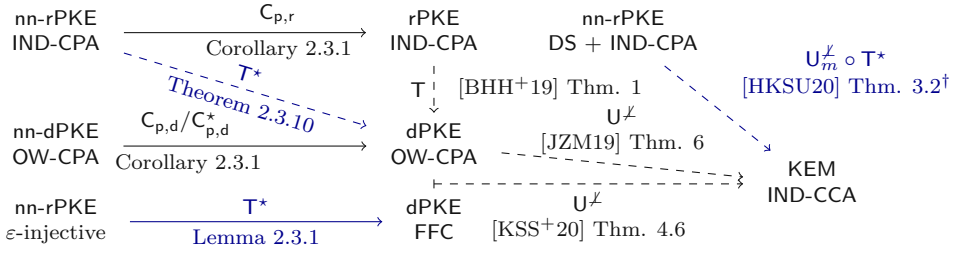

Figure 2.2: Overview of the transformations in the QROM using the notation from Figure 2.1. A dashed arrow denotes a non-tight reduction. DS denotes disjoint simulatability.
†: Obtained by applying the modifications from Theorems 2.3.9 and 2.3.10 to [HKSU20, Thm 3.2].

transform results in an increase in the sum of ciphertext and public-key size of 30% even in the best case scenario, i.e., for an IND-CPA version of KEM Round5 [GZB+19]. Nevertheless, it offers easier constant-time implementations and the opportunity of decreasing the correctness error without changing the underlying parameter set and, thus, the possibility to focus on analyzing and implementing one parameter set for both, IND-CPA and IND-CCA security.

Bloom Filter KEMs. Finally, we revisit puncturable KEMs from Bloom filter KEMs (BFKEMs) [DJSS18, DGJ+21], a recent primitive to realize 0-RTT key exchange protocols with full forward-secrecy [GHJL17]. Currently, it is unclear how to instantiate BFKEMs generically from IBE and, in particular, from conjectured post-quantum assumptions due to the correctness error of the respective IBE schemes. We show that one can construct BFKEMs generically from any IBE and even base it upon IBEs with a (non-)negligible correctness error. Consequently, our results allow BFKEMs to be instantiated from lattice- and code-based IBEs and, thereby, we obtain candidates for post-quantum CCA-secure BFKEMs.

On the progress in the NIST PQC. We note that our work has been done during the second round of the NIST PQC. Meanwhile, NIST has announced the third-round candidates[1] and finalists[2] to be standardized. From the schemes that are suitable for our compilers, BIKE [ABB+19] and FrodoKEM [NAB+19] were advanced to the third round as alternate candidates in the competition, and BIKE [ABB+19] also reached the fourth round of the competition. Moreover, we concretely analyze the submissions to the second round and want to note that meanwhile there are additional results on the cryptanalysis of some relevant second round schemes, e.g., for ROLLO in [BBC+20] as well as for LEDAcrypt in [APRS20]. These results might require a change in the parameters compared to the versions that we use in this work.

[1] https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement
[2] https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4

2.2 Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$, and let $\lambda \in \mathbb{N}$ be the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For an algorithm A, let $y \leftarrow A(\lambda, x)$ be the process of running A on input (λ, x) with access to uniformly random coins and assigning the result to y (we may assume that all algorithms take λ as input). To make the random coins r explicit, we write $A(x; r)$. We say an algorithm A is probabilistic polynomial time (PPT) if the running time of A is polynomial in λ. A function f is negligible if its absolute value is smaller than the inverse of any polynomial, i.e., if $\forall c\ \exists k_0$ s.t. $\forall \lambda \geq k_0 : |f(\lambda)| < 1/\lambda^c$.

2.2.1 Public-Key Encryption and Key-Encapsulation Mechanisms

Public-key encryption. A public-key encryption (PKE) scheme Π with message space $\mathcal{M}$ consists of the three PPT algorithms $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$: $\mathsf{KGen}(\lambda)$, on input security parameter λ, outputs public and secret keys $(\mathsf{pk}, \mathsf{sk})$. $\mathsf{Enc}(\mathsf{pk}, M)$, on input pk and message $M \in \mathcal{M}$, outputs a ciphertext ctxt. $\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$, on input sk and ctxt, outputs $M \in \mathcal{M} \cup \{\perp\}$. We may assume that pk is implicitly available in Dec.

Correctness. We recall the definition of δ-correctness of [HHK17a]. A PKE Π is δ-correct if

$$E\left[\max_{M \in \mathcal{M}} \Pr[c \leftarrow \mathsf{Enc}(\mathsf{pk}, M) : \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}) \neq M]\right] \leq \delta,$$

where the expected value is taken over all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$.

PKE-IND-CPA, PKE-OW-CPA, and PKE-OW-PCA security. We say a PKE Π is PKE-IND-CPA-secure if and only if any PPT adversary $\mathcal{A}$ has only negligible advantage in the following security experiment. First, $\mathcal{A}$ gets an honestly generated public key pk. $\mathcal{A}$ outputs equal-length messages (M_0, M_1) and, in return, gets $\mathsf{ctxt}_b^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M_b)$, for $b \leftarrow \{0, 1\}$. Eventually, $\mathcal{A}$ outputs a guess b'. If $b = b'$, then the experiment outputs 1. For PKE-OW-CPA security, $\mathcal{A}$ does not receive a ciphertext for $\mathcal{A}$-chosen messages, but only a ciphertext $\mathsf{ctxt}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$ for $M \leftarrow \mathcal{M}$ and outputs M'; if $M = M'$, then the experiment outputs 1. For PKE-OW-PCA security, $\mathcal{A}$ additionally has access to a plaintext checking oracle $\textsc{Pco}(M, \mathsf{ctxt})$ returning 1 if $M = \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$ and 0 otherwise.

Definition 2.2.1. *For any PPT adversary $\mathcal{A}$ the advantage function*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right|,$$

is negligible in λ, where the experiment $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda)$ is given in Figure 2.3 and Π is a PKE as above.

Exp. $\mathsf{Exp}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda)$	**Exp.** $\mathsf{Exp}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda)$	**Exp.** $\mathsf{Exp}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi,\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$	$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$	$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
$(M_0, M_1) \leftarrow \mathcal{A}(\mathsf{pk})$	$M \leftarrow \mathcal{M}$	$M \leftarrow \mathcal{M}$
$b \leftarrow \{0,1\}$	$\mathsf{ctxt}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$	$\mathsf{ctxt}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$
$\mathsf{ctxt}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, M_b)$	$M' \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{ctxt}^*)$	$M' \leftarrow \mathcal{A}^{\mathrm{Pco}(\cdot,\cdot)}(\mathsf{pk}, \mathsf{ctxt}^*)$
$b' \leftarrow \mathcal{A}(\mathsf{ctxt}^*)$	**if** $M = M'$ then **return** 1	**if** $M = M'$ then
if $b = b'$ then **return** 1	**else return** 0	**return** 1 **else return** 0
else return 0		

Figure 2.3: PKE-x-y security with $\mathsf{x} \in \{\mathsf{OW}, \mathsf{IND}\}$, $\mathsf{y} \in \{\mathsf{CPA}, \mathsf{PCA}\}$ for Π.

Definition 2.2.2. *For any PPT adversary $\mathcal{A}$, and $\mathsf{y} \in \{\mathsf{CPA}, \mathsf{PCA}\}$ the advantage function*

$$\mathsf{Exp}^{\mathsf{pke\text{-}OW\text{-}y}}_{\Pi,\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{pke\text{-}OW\text{-}y}}_{\Pi,\mathcal{A}}(\lambda) = 1],$$

is negligible in λ, where the experiments $\mathsf{Exp}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda)$ and $\mathsf{Exp}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi,\mathcal{A}}(\lambda)$ are given in Figure 2.3 and Π is a PKE as above.

We recall a well known lemma below:

Lemma 2.2.1. *For any adversary B there exists an adversary A with the same running time as that of B such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi,B}(\lambda) \leq \mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda) + \frac{1}{|\mathcal{M}|}.$$

We note that an analogous result to Lemma 2.2.1 holds for the ℓ-OW-CPA and ℓ-IND-CPA notions below.

Multi-challenge setting. We recall some basic observations from [BBM00] regarding the multi-challenge security of PKE schemes. In particular, for our construction we need the relation between OW-CPA/IND-CPA security in the conventional single-challenge and single-user setting and n-OW-CPA/n-IND-CPA respectively, which represents the multi-challenge and multi-user setting. In particular, latter means that the adversary is allowed to obtain multiple challenges under multiple different public keys.

Theorem 2.2.3 (Th. 4.1 [BBM00])**.** *Let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme that provides x-CPA security with $\mathsf{x} \in \{\mathsf{OW}, \mathsf{IND}\}$. Then, it holds that:*

$$\mathsf{Adv}^{\mathsf{pke\text{-}x\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda) \geq \frac{1}{q \cdot n} \cdot \mathsf{Adv}^{\mathsf{n\text{-}pke\text{-}x\text{-}cpa}}_{\Pi,\mathcal{A}}(\lambda),$$

where n is the number of public keys and $\mathcal{A}$ makes at most q queries to any of its n challenge oracles.

Exp. $\mathsf{Exp}^{\mathsf{pke\text{-}ffc}}_{\Pi,\mathcal{A}}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$

$L \leftarrow \mathcal{A}(\mathsf{pk})$

if exists $\mathsf{ctxt} \in L$ with $M \in \mathcal{M}$ such that $\mathsf{Enc}(\mathsf{pk}, M) = \mathsf{ctxt}$ and $\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}) \neq M$ **then return** 1 **else return** 0

Figure 2.4: Finding-failing-ciphertext experiment for Π.

Although the loss imposed by the reduction in Theorem 2.2.3 can be significant when used in a general multi-challenge and multi-user setting, in our application we only have cases where $n = 1$ and small q ($q = 5$ at most), or vice versa (i.e., $q = 1$ and $n = 5$ at most) thus tightness in a concrete setting is preserved.

Finding failing ciphertexts and injectivity. For the QROM security proof we will need the following two definitions from [BHH+19].

Definition 2.2.4 (ε-injectivity)**.** *A PKE Π is called ε-injective if*

- Π *is deterministic and*

$$\Pr[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda) : M \mapsto \mathsf{Enc}(\mathsf{pk}, M) \text{ is not injective}] \leq \varepsilon.$$

- Π *is non-deterministic with randomness space $\mathcal{R}$ and*

$$\Pr\left[\begin{matrix}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda), \\ M, M' \leftarrow \mathcal{M}, r, r' \leftarrow \mathcal{R}\end{matrix} : \mathsf{Enc}(\mathsf{pk}, M; r) = \mathsf{Enc}(\mathsf{pk}, M'; r')\right] \leq \varepsilon.$$

Definition 2.2.5 (Finding failing ciphertexts)**.** *For a deterministic* PKE*, the* FFC*-advantage of an adversary $\mathcal{A}$ is defined as*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ffc}}_{\Pi,\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{pke\text{-}ffc}}_{\Pi,\mathcal{A}}(\lambda) = 1],$$

where the experiment $\mathsf{Exp}^{\mathsf{pke\text{-}ffc}}_{\Pi,\mathcal{A}}$ is given in Figure 2.4.

Key-encapsulation mechanism. A key-encapsulation mechanism (KEM) scheme KEM with key space $\mathcal{K}$ consists of the three PPT algorithms $(\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Decaps})$: $\mathsf{KGen}(\lambda)$, on input security parameter λ, outputs public and secret keys $(\mathsf{pk}, \mathsf{sk})$. $\mathsf{Encaps}(\mathsf{pk})$, on input pk, outputs a ciphertext ctxt and key k. $\mathsf{Decaps}(\mathsf{sk}, \mathsf{ctxt})$, on input sk and ctxt, outputs k or $\{\perp\}$.

Correctness of KEM. We call a KEM δ-correct if for all $\lambda \in \mathbb{N}$, for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$, for all $(\mathsf{ctxt}, \mathsf{k}) \leftarrow \mathsf{Enc}(\mathsf{pk})$, we have that

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}) \neq \mathsf{k}] \leq \delta.$$

Exp. $\mathsf{Exp}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
$(\mathsf{ctxt}^*, \mathsf{k}_0) \leftarrow \mathsf{Encaps}(\mathsf{pk}), \mathsf{k}_1 \leftarrow \mathcal{K}$
$b \leftarrow \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathsf{Decaps}(\mathsf{sk},\cdot)}(\mathsf{pk}, \mathsf{ctxt}^*, \mathsf{k}_b)$
if $b = b'$ **then return** 1 **else return** 0

Figure 2.5: KEM-IND-CCA security experiment for KEM.

KEM-IND-CCA security. We say a KEM KEM is KEM-IND-CCA-secure if and only if any PPT adversary $\mathcal{A}$ has only negligible advantage in the following security experiment. First, $\mathcal{A}$ gets an honestly generated public key pk as well as a ciphertext-key pair $(\mathsf{ctxt}^*, \mathsf{k}_b)$, for $(\mathsf{ctxt}^*, \mathsf{k}_0) \leftarrow \mathsf{Encaps}(\mathsf{pk})$, for $\mathsf{k}_1 \leftarrow \mathcal{K}$, and for $b \leftarrow \{0,1\}$. $\mathcal{A}$ has access to a decapsulation oracle $\mathsf{Dec}(\mathsf{sk},\cdot)$ and we require that $\mathcal{A}$ never queries $\mathsf{Decaps}(\mathsf{sk}, \mathsf{ctxt}^*)$. Eventually, $\mathcal{A}$ outputs a guess b'. Finally, if $b = b'$, then the experiment outputs 1.

Definition 2.2.6. *For any PPT adversary $\mathcal{A}$, the advantage functions*

$$\mathsf{Adv}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},\mathcal{A}}(\lambda) := \left| \Pr[\mathsf{Exp}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|,$$

is negligible in λ, where the experiment $\mathsf{Exp}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ is given in Figure 2.5 and KEM *is a KEM as above.*

2.2.2 Identity-Based Encryption

An identity-based encryption (IBE) scheme IBE with identity space $\mathcal{ID}$ and message space $\mathcal{M}$ consists of the PPT algorithms $(\mathsf{KGen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$: $\mathsf{KGen}(\lambda)$ on input security parameter λ, outputs main public and secret keys $(\mathsf{mpk}, \mathsf{msk})$. $\mathsf{Ext}(\mathsf{msk}, id)$ on input identity $id \in \mathcal{ID}$, outputs an identity secret key sk_{id}. $\mathsf{Enc}(\mathsf{mpk}, id, M)$ on input mpk, $id \in \mathcal{ID}$, and message $M \in \mathcal{M}$, outputs a ciphertext ctxt. $\mathsf{Dec}(\mathsf{sk}_{id}, \mathsf{ctxt})$ on input sk_{id} and ctxt, outputs $M \in \mathcal{M} \cup \{\perp\}$.

Correctness of IBE. Analogous to section 2.3, we say that an IBE IBE is

- $\delta(\cdot)$-correct if for any $id \in \mathcal{ID}$ and all $M \in \mathcal{M}$:

$$\Pr\left[\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{mpk}, id, M) : \mathsf{Dec}(\mathsf{sk}_{id}, \mathsf{ctxt}) \neq M\right] \leq \delta(\lambda),$$

  where the probability is taken over the random coins of the encryption algorithm, $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{KGen}(\lambda)$, and $\mathsf{sk}_{id} \leftarrow \mathsf{Ext}(\mathsf{msk}, id)$.

- $\epsilon(\cdot)$-key $\delta(\cdot)$-correct if for any $id \in \mathcal{ID}$ and $M \in \mathcal{M}$: except with probability at most $\epsilon(\lambda)$, key pairs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{KGen}(\lambda)$ are such that

$$\Pr\left[\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{mpk}, id, M) : \mathsf{Dec}(\mathsf{sk}_{id}, \mathsf{ctxt}) \neq M\right] \leq \delta(\lambda),$$

where the probability is taken over the random coins (possibly obtained from a random oracle) of the encryption algorithm.

IBE-sIND-CPA security of IBE. We say an IBE scheme IBE is IBE-sIND-CPA-secure if and only if any PPT adversary $\mathcal{A}$ has only negligible advantage in the following security experiment. First, $\mathcal{A}$ outputs the target identity id^* and, subsequently, gets an honestly generated main public key mpk. During the experiment, but after providing id^*, $\mathcal{A}$ has access to a secret-key extraction oracle $\mathsf{Ext}(\mathsf{msk}, \cdot)$ where we require that $\mathcal{A}$ never queries an identity secret key for id^*. At some point, $\mathcal{A}$ outputs equal-length messages (M_0, M_1) and receives a challenge ciphertext $\mathsf{ctxt}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, id^*, M_b)$, for $b \leftarrow \{0,1\}$. Eventually, $\mathcal{A}$ outputs a guess b'; if $b = b'$, then the experiment outputs 1. The experiment is depicted in Figure 2.6.

Definition 2.2.7. *For any PPT adversary $\mathcal{A}$, the advantage function*

$$\mathsf{Adv}^{\mathsf{ibe\text{-}sind\text{-}cpa}}_{\mathsf{IBE},B}(\lambda) := \left| \Pr[\mathsf{Exp}^{\mathsf{ibe\text{-}sind\text{-}cpa}}_{\mathsf{IBE},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|,$$

is negligible in λ, where the experiment $\mathsf{Exp}^{\mathsf{ibe\text{-}sind\text{-}cpa}}_{\mathsf{IBE},\mathcal{A}}(\lambda)$ is given in Figure 2.6 and IBE *is an IBE scheme.*

Exp. $\mathsf{Exp}^{\mathsf{ibe\text{-}sind\text{-}cpa}}_{\mathsf{IBE},\mathcal{A}}(\lambda)$

$id^* \leftarrow \mathcal{A}(\lambda)$

$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{KGen}(\lambda)$

$(M_0, M_1) \leftarrow \mathcal{A}^{\mathsf{Ext}(\mathsf{msk},\cdot)}(\mathsf{mpk})$

$b \leftarrow \{0,1\}$

$\mathsf{ctxt}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, id^*, M_b)$

$b' \leftarrow \mathcal{A}^{\mathsf{Ext}(\mathsf{msk},\cdot)}(\mathsf{ctxt}^*)$

if $b = b'$ then return 1 else return 0

Figure 2.6: IBE-sIND-CPA experiment for IBE scheme IBE.

γ-spreadness of IBE. In order to prove our Bloom filter KEM CCA-secure in Section 2.5, we need an additionally property of the underlying IBE scheme which essentially guarantees that honestly generated IBE ciphertexts have large-enough min-entropy.

Definition 2.2.8 (γ-Spreadness of IBE)**.** *For all $\lambda \in \mathbb{N}$, an IBE scheme* IBE *is* γ-spread, *if for any $(\mathsf{mpk}, \cdot) \leftarrow \mathsf{KGen}(\lambda)$, any identity $id \in \mathcal{ID}$, any message $M \in \mathcal{M}$, any $C \in \mathcal{C}$, and $r \leftarrow \mathcal{R}$, where $\mathcal{C}$ and $\mathcal{R}$ are the ciphertext and randomness spaces of* IBE*, respectively, we have that $\Pr[C = \mathsf{Enc}(\mathsf{mpk}, id, M; r)] \leq 2^{-\gamma}$ holds, where the probability is taken over the random coins of* KGen.

2.3 CCA Security from Non-Negligible Correctness Errors

In this section, we present our approaches to generically achieve CCA secure KEMs in the (Q)ROM with negligible correctness error when starting from an OW-CPA or IND-CPA secure PKE with non-negligible correctness error. We start by discussing the definitions of correctness errors of PKE and KEMs. Then, we present a generic transform based on the direct product compiler of Dwork et al. [DNR04] and revisit certain FO transformation variants from [HHK17a] (in particular the T and U transformations), their considerations in the QROM [BHH+19] and their application with the direct product compiler. As a better alternative, we analyze the non-black-box use of the previous technique yielding transformation $\mathsf{T}^\star$, that combines the direct product compiler with the T transformation. Finally, we provide a comprehensive comparison of the two approaches.

2.3.1 On the Correctness Error

In this work, we use δ-correctness definitions for PKEs slightly derived from that given by HHK in [HHK17a]. These definitions are tailored, as done in [HHK17a] via maxing over all possible messages, to the security proofs of the FO-transforms where an adversary could actively search for the worst possible message, in order to trigger decryption failure. Moreover, they are also tailored, via taking the probability over appropriately chosen random coins, to be compatible with the $\mathsf{C}^\star_{\mathsf{p,d}}$ and $\mathsf{T}^\star$ transformation respectively. As done by Dwork et al. [DNR04], we explicitly write the correctness error as a function in the security parameter:

Definition 2.3.1. *A* PKE Π *is*

- $\delta(\cdot)$*-correct if for all* $M \in \mathcal{M}$*:*

$$\Pr[\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pk}, M) : \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}) \neq M] \leq \delta(\lambda),$$

  where the probability is taken over the random coins of the encryption algorithm and that of $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$*.*

- $\epsilon(\cdot)$*-key* $\delta(\cdot)$*-correct if for all* $M \in \mathcal{M}$*: except that with probability at most* $\epsilon(\lambda)$*, key pairs* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$ *are such that*

$$\Pr[\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pk}, M) : \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}) \neq M] \leq \delta(\lambda),$$

  where the probability is taken over the random coins (possibly obtained from a random oracle) of the encryption algorithm.

Remark 1. *In the rest of the paper, we will use the first definition,* $\delta(\cdot)$*-correctness, for (truly) deterministic PKEs (to which the* $\mathsf{C}^\star_{\mathsf{p,d}}$ *compiler is applied), and the second definition,* $\epsilon(\cdot)$*-key* $\delta(\cdot)$*-correctness, for randomized and derandomized PKEs (which are dealt with via the* $\mathsf{T}^\star$ *transformation). With this distinction in mind, in both cases, for ease of exposition, we will sometime call it simply correctness error.*

Exp. $\mathsf{Exp}^{\mathsf{cor}}_{\Pi,\mathcal{A}}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$
$M \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{sk})$
if $M \neq \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M))$ **then return** 1 **else return** 0

Figure 2.7: Correctness experiment for PKE.

It will be important for our transform to make explicit that the correctness error depends on the security level. We will often just write $\delta = \delta(\lambda)$. If Π is defined relative to a random oracle H, then the adversary is given access to the random oracle and δ is additionally a function in the number of queries q_{H}, i.e., the bound is given by $\leq \delta(\lambda, q_{\mathsf{H}})$.

We note that in [BS20] an alternative definition of correctness was proposed, where the adversary does not get access to sk and the adversary's runtime is bounded. With this change, it can be run as part of the IND-CCA experiment which does not change the power of the IND-CCA adversary and additionaly removes a factor q_{H} from the correctness error and advantage analysis. In particular, one can obtain an upper bound for IND-CCA security of a scheme via the correctness error.

We recall, for completeness, the definitions of correctness error, here denoted as HHK-δ-correctness (from Hofheinz-Hövelmanns-Kiltz) and DNR-δ-correctness (from Dwork-Naor-Reingold), used by Hofheinz et al. and Dwork et al. respectively:

Definition 2.3.2 (Sect. 2.1 [HHK17a])**.** *A* PKE Π *is HHK-$\delta(\cdot)$-correct if for all $M \in \mathcal{M}$:*

$$E\left[\max_{m \in \mathcal{M}} \Pr[\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pk}, M) : \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}) \neq M]\right] \leq \delta(\lambda),$$

where the expected value is taken over all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$.

With this definition, particularly bad keys in terms of correctness error only contribute a fraction to the overall correctness error as it averages the error probability over all key pairs. An alternative but equivalent definition, as used in [HHK17a], can be given in the following form: a PKE Π is called HHK-$\delta(\cdot)$-correct if we have for all (possibly unbounded) adversaries $\mathcal{A}$ that

$$\mathsf{Adv}^{\mathsf{cor}}_{\Pi,\mathcal{A}}(\lambda) = \Pr[\mathsf{Exp}^{\mathsf{cor}}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq \delta(\lambda),$$

where the experiment is given in Figure 2.7.

Definition 2.3.3 (Def. 2, Def. 3 [DNR04])**.** *A* PKE Π *is*

- *DNR-$\delta(\cdot)$-correct if we have that*

  $$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M)) \neq M] \leq \delta(\lambda),$$

  where the probability is taken over the choice of key pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$, $M \in \mathcal{M}$ *and over the random coins of* Enc *and* Dec.

$\Pi'.\mathsf{KGen}'(\lambda, \ell)$

// if $\mathsf{C}_{\mathsf{p,r}}$
return $\Pi.\mathsf{KGen}(\lambda)$
// if $\mathsf{C}_{\mathsf{p,d}}$
for $i \in [\ell]$
$(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \Pi.\mathsf{KGen}(\lambda)$
$\mathsf{pk} := (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$
$\mathsf{sk} := (\mathsf{sk}_1, \ldots, \mathsf{sk}_\ell)$
return $(\mathsf{pk}, \mathsf{sk})$

$\Pi'.\mathsf{Enc}'(\mathsf{pk}, M)$

for $i \in [\ell]$
// if $\mathsf{C}_{\mathsf{p,r}}$
$r_i \leftarrow \Pi.\mathcal{R}$
$\mathsf{ctxt}_i \leftarrow \Pi.\mathsf{Enc}(\mathsf{pk}, M; r_i)$
// if $\mathsf{C}_{\mathsf{p,d}}$
$\mathsf{ctxt}_i \leftarrow \Pi.\mathsf{Enc}(\mathsf{pk}_i, M)$
$\mathsf{ctxt} := (\mathsf{ctxt}_1 \ldots, \mathsf{ctxt}_\ell)$
return ctxt

$\Pi'.\mathsf{Dec}'(\mathsf{sk}, \mathsf{ctxt})$

$\mathsf{ctxt} := (\mathsf{ctxt}_1 \ldots, \mathsf{ctxt}_\ell)$
for $i \in [\ell]$
// if $\mathsf{C}_{\mathsf{p,r}}$
$M'_i := \Pi.\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}_i)$
// if $\mathsf{C}_{\mathsf{p,d}}$
$M'_i := \Pi.\mathsf{Dec}(\mathsf{sk}_i, \mathsf{ctxt}_i)$
return $\mathsf{maj}(M'_1, \ldots, M'_\ell)$

Figure 2.8: Compilers $\mathsf{C}_{\mathsf{p,d}}$ and $\mathsf{C}_{\mathsf{p,r}}$.

- *DNR-(almost-)all-keys $\delta(\cdot)$-correct if for all (but negligible many) keys* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$*, we have that*

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M)) \neq M] \leq \delta(\lambda),$$

where the probability is taken over the choice of $M \in \mathcal{M}$ and over the random coins of Enc *and* Dec*.*

Correctness error in this sense still allows bad key pairs that potentially have an even worse error but it is not suited for our security proofs as the probability is also taken over $M \leftarrow \mathcal{M}$. Recently Drucker et al. [DGKP21] introduced the notion of message agnostic PKE and showed that all the versions of BIKE, a 2nd round candidate in the NIST PQC, are message-agnostic: in such a PKE, the probability that, given $(\mathsf{sk}, \mathsf{pk})$, the encryption of a message $M \in \mathcal{M}$ correctly decrypts is independent of the message $M \in \mathcal{M}$ itself. For such PKEs the definitions of δ-correctness and DNR-δ-correctness coincide (Cor. 1 [DGKP21]).

2.3.2 Compiler for Immunizing Decryption Errors

Now we present two variants of a compiler C_{p} denoted $\mathsf{C}_{\mathsf{p,d}}$ (for deterministic schemes) and $\mathsf{C}_{\mathsf{p,r}}$ (for randomized schemes) which is based on the direct product compiler by Dwork et al. [DNR04]. We recall that the idea is to take a PKE scheme $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ with non-negligible correctness error δ (and randomness space $\mathcal{R}$ in case of randomized schemes) and output a PKE scheme $\Pi' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ with negligible correctness error δ' (and randomness space $\mathcal{R}' := \mathcal{R}^\ell$, for some $\ell \in \mathbb{N}$, in case of a randomized schemes). We present a precise description of the compilers in Figure 2.8. Note that in Dec', the message that is returned most often by Dec is returned. If two or more messages are tied, one of them is returned arbitrarily and we denote this operation as $\mathsf{maj}(M')$.

Analyzing correctness. Dwork et al. in [DNR04] explicitly discuss the amplification of the correctness for encryption schemes with a binary message space $\mathcal{M} = \{0, 1\}$ and

obtain that to achieve DNR-δ'-correctness $\ell > \frac{c}{(1-\delta)^2} \cdot \log \frac{1}{\delta'}$ when starting from a scheme with DNR-δ-correctness. As c is some constant that is never made explicit, the formula is more of theoretical interest and for concrete instances it is hard to estimate the number of required ciphertexts. We can however analyze the probabilities that the majority vote in Dec' returns the correct result. As far as the correctness notion used in this work is concerned, in order to prove an acceptable good lower bound for the δ-correctness of the direct product compiler, it suffices to find an event, in which the decryption procedure fails, that happens with a large enough probability. The following reasoning applies to both its deterministic and randomized versions, $\mathsf{C}_{\mathsf{p,d}}$ and $\mathsf{C}_{\mathsf{p,r}}$ respectively. One such case is the following: only 1 ciphertext correctly decrypts and all other $\ell - 1$ ciphertexts decrypt to $\ell - 1$ distinct wrong messages. During the maj operation, one of the "wrong" messages is then returned. The probability of this event is

$$\frac{\ell - 1}{\ell} \cdot \binom{\ell}{\ell - 1} \cdot \delta^{\ell-1} \cdot (1 - \delta) \cdot \frac{M - 1}{M - 1} \cdot \frac{M - 2}{M - 1} \cdots \frac{M - (\ell - 1)}{M - 1}.$$

Looking ahead to our compiler T^* presented in Section 2.3.4, if the message space is sufficiently large, this probability is bigger than $\delta^{\ell-1}(1 - \delta)$, which gives that at least one more ciphertext is needed to achieve the same decryption error as with our compiler T^*. The results are shown in Table 2.1. One can compute the exact probability of decryption error by listing all cases in which the decryption fails and summing up all these probabilities to obtain the overall decryption failure of the direct product compiler. This computation is not going to give a significantly different result from the lower bound that we have just computed.

We note that using 2 parallel ciphertexts does not improve the correctness error, so the direct product compiler only becomes interesting for $\ell \geq 3$: indeed for $\ell = 2$, we have 3 possible outcomes in which the decryption algorithm can fail: 1) the first ciphertext decrypts and the second does not, 2) vice versa, 3) both fail to decrypt. In 1), 2), half the time the wrong plaintext is returned. Summing these probabilities gives exactly δ.

Remark 2. *As far as the deterministic direct product compiler $\mathsf{C}_{\mathsf{p,d}}$ is concerned, the correctness error can be improved by modifying the decryption: instead of relying on the maj operation, we can re-encrypt the plaintexts obtained during decryption with the respective keys and compare them to the original ciphertexts. Only if this check passes, the plaintext is returned. If this is done, then decryption fails with probability $\ell\delta^\ell$ and thereby the number of parallel repetition necessary to achieve negligible correctness-error is reduced at the cost of a computational overhead in the decryption. We denote this version of the deterministic direct product compiler by $\mathsf{C}^{\star}_{\mathsf{p,d}}$. The bound $\ell\delta^\ell$ on the correctness error can be derived as follows: the wrong message is returned if*

- *no ciphertext component decrypts correctly, or*
- *at least one ciphertext component correctly decrypts but a wrong message is anyway returned.*

Table 2.1: Estimation of the correctness error for the direct product compilers. $\delta'(\ell)$ denotes the correctness error for ℓ ciphertexts.

δ	$\delta'(2)$	$\delta'(3)$	$\delta'(4)$
2^{-32}	$\approx 2^{-32}$	$\approx 2^{-63}$	$\approx 2^{-94}$
2^{-64}	$\approx 2^{-64}$	$\approx 2^{-127}$	$\approx 2^{-190}$
2^{-96}	$\approx 2^{-96}$	$\approx 2^{-191}$	$\approx 2^{-284}$

To analyze the probabilities of such events happening, we start by defining a define a tuple $((\mathsf{pk}, \mathsf{sk}), M)$ problematic*, if it exhibits a correctness error in* Π*, i.e.,* $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M)) \neq M$*. By definition of* δ*-correctness, each tuple is* problematic *with probability at most* δ*, as* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$ *outputs independently random key pairs.*

Let us consider the first event. The probability of the first event happening equals the probability of each $((\mathsf{pk}_j, \mathsf{sk}_j), M)$*,* $j \in [\ell]$*, being* problematic*. Since* KeyGen *outputs are independent, this can be bounded by* δ^ℓ*.*

Let us now consider the second event. Suppose message M *was encrypted but message* $M' \neq M$ *gets decrypted in some slot* $i \in [\ell]$*, such a message passes all checks and gets returned. Since* $C_i = \mathsf{Enc}(\mathsf{pk}_i, M)$ *but* $\Pi.\mathsf{Dec}(\mathsf{sk}_i, C_i) = M'$*, we deduce that the tuple* $((\mathsf{pk}_i, \mathsf{sk}_i), M)$ *is* problematic*. Moreover, since* M' *passes all re-encryption checks, which in particular means that for all* $j \in [\ell] \setminus \{i\}$

$$\mathsf{Enc}(\mathsf{pk}_j, M) = C_j = \mathsf{Enc}(\mathsf{pk}_j, M').$$

Since $\Pi.\mathsf{Dec}$ *is deterministic, at most one between* M *and* M' *can be equal to* $\Pi.\mathsf{Dec}(\mathsf{sk}_j, C_j)$*. Therefore, either* $((\mathsf{pk}_j, \mathsf{sk}_j), M')$ *is* problematic *or* $((\mathsf{pk}_j, \mathsf{sk}_j), M)$ *is* problematic*. As we have remarked before, each such tuple is* problematic *with probability at most* δ*. Thus, the overall probability of* M' *getting returned is* δ^ℓ*. Since there are* $\ell - 1$ *such possible indices* i *(recall that in the second event at least one ciphertext component correctly decrypts), a union bound shows that the probability of this second event happening is bounded by* $(\ell - 1)\delta^\ell$*.*

Putting everything together, we obtain that Π' *has then correctness error* $\delta' := \delta^\ell + (\ell - 1)\delta^\ell = \ell\delta^\ell$*.*

Their security follows by applying Theorem 2.2.3 with $q = 1$ and $n = \ell$ in the deterministic case, for both $\mathsf{C}_{\mathsf{p,d}}$ and $\mathsf{C}_{\mathsf{p,d}}{}^\star$, or vice versa with $q = \ell$ and $n = 1$ in the randomized case:

Corollary 2.3.1. *For any* x-CPA *adversary* B *against* Π' *obtained via applying* $\mathsf{C}_{\mathsf{p,y}}$ *to* Π*, there exists an* x-CPA *adversary* A *such that:*

$$\mathsf{Adv}^{\mathsf{pke\text{-}x\text{-}cpa}}_{\Pi',B}(\lambda) \leq \ell \cdot \mathsf{Adv}^{\mathsf{pke\text{-}x\text{-}cpa}}_{\Pi,A}(\lambda),$$

where $\mathsf{y} = \mathsf{d}$ *if* $\mathsf{x} = \mathsf{OW}$ *and* $\mathsf{y} = \mathsf{r}$ *if* $\mathsf{x} = \mathsf{IND}$*.*

$\Pi'.\mathsf{Enc}(\mathsf{pk}, M)$	$\Pi'.\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$
$C := \Pi.\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M))$	$M' := \Pi.\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$
return ctxt	**if** $M' = \bot$ **or** $\mathsf{ctxt} \neq \Pi.\mathsf{Enc}(\mathsf{pk}, M'; \mathsf{G}(M'))$
	return $\bot$
	else return M'

Figure 2.9: OW-PCA-secure scheme $\Pi' = \mathsf{T}[\Pi, \mathsf{G}]$ with deterministic encryption.

As the analysis above suggests, ℓ will be a small constant, so the loss in ℓ does not pose a problem regarding tightness.

2.3.3 Transformations T and $\mathsf{U}^{\not\perp}$

Subsequently, we discuss basic transformations from [HHK17a] to first transform an IND-CPA secure PKE into an OW-CPA secure PKE (transformation T in [HHK17a]) and then to convert an OW-PCA secure PKE into an IND-CCA secure KEM with implicit rejection (transformation $\mathsf{U}^{\not\perp}$ in [HHK17a]) and we discuss alternative transformations later. We stress that these transformations either work for perfectly correct schemes or schemes with a negligible correctness error.

T: IND-CPA $\implies$ OW-PCA (ROM)/OW-CPA (QROM). The transform T is a simple de-randomization of a PKE by deriving the randomness r used by the algorithm Enc via evaluating a random oracle (RO) on the message to be encrypted. More precisely, let $\Pi = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE with message space $\mathcal{M}$ and randomness space $\mathcal{R}$ and $\mathsf{G}\colon \mathcal{M} \to \mathcal{R}$ be a RO. We denote the PKE Π' obtained by applying transformation T depicted in Figure 2.9 as $\Pi' = \mathsf{T}[\Pi, \mathsf{G}]$, where $\Pi'.\mathsf{KGen} = \Pi.\mathsf{KGen}$ and is thus omitted.

For the ROM, we recall the following theorem:

Theorem 2.3.4 (Thm. 3.2 [HHK17a] (Π IND-CPA $\implies$ Π' OW-PCA))**.** *Assume Π to be δ-correct. Then, Π' is $\delta_1(q_\mathsf{G}) = q_\mathsf{G} \cdot \delta$ correct and for any* OW-PCA *adversary B that issues at most q_G queries to the RO* G *and q_P queries to a plaintext checking oracle $\textsc{Pco}$, there exists an* IND-CPA *adversary A running in about the same time as B such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi',B}(\lambda) \leq (q_\mathsf{G} + q_P) \cdot \delta + \frac{2q_\mathsf{G} + 1}{|\mathcal{M}|} + 3 \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda).$$

And for the QROM, we recall the following theorem:

Theorem 2.3.5 (Thm. 1 [BHH+19] (Π IND-CPA $\implies$ Π' OW-CPA))**.** *If $\mathcal{A}$ is an* OW-CPA*-adversary against $\Pi' = \mathsf{T}[\Pi, \mathsf{G}]$ issuing at most q_G queries to the quantum-accessible RO* G *of at most depth d, then there exists an* IND-CPA *adversary B against Π running in about the same time as $\mathcal{A}$ such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi',\mathcal{A}}(\lambda) \leq (d+1) \cdot \left(\mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,B}(\lambda) + \frac{8 \cdot (q_\mathsf{G} + 1)}{|\mathcal{M}|} \right).$$

$\underline{\mathsf{KEM.KGen}(\lambda)}$	$\underline{\mathsf{KEM.Encaps}(\mathsf{pk})}$	$\underline{\mathsf{KEM.Decaps}(\mathsf{sk}, \mathsf{ctxt})}$
$(\mathsf{pk}', \mathsf{sk}') \leftarrow \Pi'.\mathsf{KGen}(\lambda)$	$M \leftarrow \mathcal{M}$	Parse $\mathsf{sk} = (\mathsf{sk}', s)$
$s \leftarrow \mathcal{M}$	$C \leftarrow \Pi'.\mathsf{Enc}(\mathsf{pk}, M)$	$M' := \Pi'.\mathsf{Dec}(\mathsf{sk}', \mathsf{ctxt})$
$\mathsf{sk} := (\mathsf{sk}', s)$	$K := \mathsf{H}(M, \mathsf{ctxt})$	**if** $M' \neq \perp$
return $(\mathsf{pk}', \mathsf{sk})$	**return** (K, ctxt)	**return** $K := \mathsf{H}(M', \mathsf{ctxt})$
		else return $K := \mathsf{H}(s, \mathsf{ctxt})$

Figure 2.10: IND-CCA-secure KEM scheme $\mathsf{KEM} = \mathsf{U}^{\not\perp}[\Pi', \mathsf{H}]$.

$\mathsf{U}^{\not\perp}$: OW-PCA $\implies$ IND-CCA. The transformation $\mathsf{U}^{\not\perp}$ transforms any OW-PCA secure PKE Π' into an IND-CCA secure KEM in the (Q)ROM. The basic idea is that one encrypts a random message M from the message space $\mathcal{M}$ of Π' and the encapsulated key is the RO evaluated on the message M and the corresponding ciphertext ctxt under Π'. This transformation uses implicit rejection and on decryption failure does not return $\perp$, but an evaluation of the RO on the ciphertext and a random message $s \in \mathcal{M}$, being part of sk of the resulting KEM, as a "wrong" encapsulation key. It is depicted in Figure 2.10.

In the ROM, we have the following result:

Theorem 2.3.6 (Thm. 3.4 [HHK17a] (Π' OW-PCA $\implies$ KEM IND-CCA))**.** *If Π' is δ_1-correct, then* KEM *is δ_1-correct in the random oracle model. For any* IND-CCA *adversary B against* KEM*, issuing at most q_{H} queries to the random oracle* H*, there exists an* OW-PCA *adversary A against Π' running in about the same time as B that makes at most q_{H} queries to the PCO oracle such that*

$$\mathsf{Adv}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{KEM},B}(\lambda) \leq \frac{q_{\mathsf{H}}}{|\mathcal{M}|} + \mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi',A}(\lambda).$$

For the QROM, we have the following non-tight result:

Theorem 2.3.7 (Thm. 6 [JZM19] (Π' OW-PCA $\implies$ KEM IND-CCA))**.** *Let Π' be a deterministic PKE scheme which is independent of* H*. Let B be an* IND-CCA *adversary against the KEM $\mathsf{U}^{\not\perp}[\Pi', \mathsf{H}]$, and suppose that $\mathcal{A}$ makes at most q_d (classical) decryption queries and q_{H} queries to quantum-accessible random oracle* H *of depth at most d, then there exists and adversary B against Π' such that*

$$\mathsf{Adv}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{U}^{\not\perp}[\Pi',\mathsf{H}],\mathcal{A}}(\lambda) \leq \frac{2 \cdot q_{\mathsf{H}}}{\sqrt{|\mathcal{M}|}} + 2 \cdot \sqrt{(q_{\mathsf{H}}+1) \cdot (2 \cdot \delta + \mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi',B}(\lambda))}.$$

If we assume ε-injectivity and FFC, respectively, we have tighter bounds:

Theorem 2.3.8 (Thm. 4.6 [KSS+20] (Π' OW-CPA + FFC $\implies$ KEM IND-CCA))**.** *Let Π' be an ε-injective deterministic PKE scheme which is independent of* H*. Suppose that $\mathcal{A}$ is an* IND-CCA *adversary against the KEM $\mathsf{U}^{\not\perp}[\Pi', \mathsf{H}]$, and suppose that $\mathcal{A}$ makes at most q_d (classical) decryption queries and q_{H} queries to quantum-accessible random oracle* H *of depth at most d, then there exist two adversaries running in about the same time as $\mathcal{A}$:*

- *an* OW-CPA*-adversary* B_1 *against* Π' *and*
- *a* FFC*-adversary* B_2 *against* Π' *returning a list of at most* q_d *ciphertexts,*

such that

$$\mathsf{Adv}^{\mathsf{kem\text{-}ind\text{-}cca}}_{\mathsf{U}^{\not\perp}[\Pi',\mathsf{H}],\mathcal{A}}(\lambda) \leq 4d \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi',B_1}(\lambda) + 6 \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ffc}}_{\Pi',B_2}(\lambda) + (4 \cdot d + 6) \cdot \varepsilon.$$

$\mathsf{FO}^{\not\perp}[\Pi, \mathsf{G}, \mathsf{H}]$**.** By combining transformation T with $\mathsf{U}^{\not\perp}$ one consequently obtains an IND-CCA secure KEM KEM from an IND-CPA secure PKE Π. Note that the security reduction of the $\mathsf{FO}^{\not\perp} := \mathsf{U}^{\not\perp} \circ \mathsf{T}$ variant of the FO is tight in the random oracle model and works even if Π has negligible correctness error instead of perfect correctness.

$\mathsf{FO}^{\not\perp}[\Pi, \mathsf{G}, \mathsf{H}]$ **in the QROM.** Hofheinz et al. in [HHK17a] also provide variants of the FO transform that are secure in the QROM, but they are (highly) non-tight. Bindel et al. [BHH+19] presented a tighter proof for $\mathsf{U}^{\not\perp}$ under an additional assumption of ε-injectivity. This result was recently improved by Kuchta et al. [KSS+20]. Additionally, Jiang et al. [JZM19] provided tighter proofs for the general case.

$\mathsf{U}^{\perp}$**,** $\mathsf{U}^{\perp}_m$**,** $\mathsf{U}^{\not\perp}_m$ **and other approaches.** Besides the transform with implicit rejection, $\mathsf{U}^{\not\perp}$, one can also consider explicit rejection, $\mathsf{U}^{\perp}$ and versions of both where the derived session key depends on the ciphertext, $\mathsf{U}^{\not\perp}_m$ and $\mathsf{U}^{\perp}_m$, respectively. Bindel et al. [BHH+19] show that security of implicit rejection implies security with explicit rejection. The opposite direction also holds if the scheme with explicit rejection also employs key confirmation. Moreover, they show that the security is independent of including the ciphertext in the session key derivation.

A different approach was proposed by Saito et al. [SXY18], where they start from a deterministic disjoint simulatable PKE and apply $\mathsf{U}^{\not\perp}_m$ with an additional re-encryption step in the decryption algorithm. While the original construction relied on a perfectly correct PKE, Jiang et al. gave non-tight reductions for schemes with negligible correctness error in [JZC+18]. Hövelmanns et al. [HKSU20] improve over this approach by giving a different modularization of Saito et al.'s TPunc.

Black-box use of the compiler $\mathsf{C}_{\mathsf{p,d}}/\mathsf{C}_{\mathsf{p,d}}{}^{\star}/\mathsf{C}_{\mathsf{p,r}}$**.** Using $\mathsf{C}_{\mathsf{p,d}}$, $\mathsf{C}_{\mathsf{p,d}}{}^{\star}$ or $\mathsf{C}_{\mathsf{p,r}}$ from Section 2.3.2, we can transform any deterministic or randomized PKE with non-negligible correctness error into one with negligible correctness error. Consequently, Theorem 2.3.1 as a result yields a scheme that is compatible with all the results on the T and variants of the U transformations in this section. Note that in particular this gives us a general way to apply these variants of the FO transform to PKE schemes with non-negligible correctness error.

2.3.4 Non Black-Box Use: the Transformation $\mathsf{T}^\star$

Since the direct product compiler is rather complicated to analyze, we alternatively investigate to start from an IND-CPA secure PKE Π which is ϵ-key δ-correct, for some non-negligible δ and introduce a variant of the transform T to de-randomize a PKE, denoted $\mathsf{T}^\star$. The idea is that we compute ℓ independent encryptions of the same message M under the same public key pk using randomness $\mathsf{G}(M, i)$, $i \in [\ell]$, where G is a RO (see Figure 2.11 for a compact description).

The resulting de-randomized PKE Π' is ϵ'-key δ'-correct, with $\epsilon' := \epsilon$ and $\delta' := \ell\delta^\ell$. The reasoning is as follows: let Bad be the event where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$ is one of those key pairs not satisfying the δ-correctness bound. Conditioned on $\neg\mathsf{Bad}$, we will show that the probability of obtaining a decryption failure is $\ell\delta^\ell$. This will prove our claim. Indeed, for any key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$ satisfying the δ-correctness bound, a decryption error occurs if one of the following two (disjoint) events happens

- no ciphertext component decrypts correctly, or
- at least one ciphertext component correctly decrypts but a wrong message is anyway returned.

Similar to what was done before, to analyze the probabilities of such events happening, we start by defining a query $\mathsf{G}(M, i)$ *problematic* iff it exhibits a correctness error in Π (in the sense that $\Pi.\mathsf{Dec}(\mathsf{sk}, \Pi.\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M, i))) \neq M$). By definition, each query $\mathsf{G}(M, i)$ is *problematic* with probability at most δ, as G outputs independently random values.

Let us consider the first event. The probability of the first event happening equals the probability of each $\mathsf{G}(M, j)$, $j \in [\ell]$, being *problematic*. Since G's outputs are independent, this can be bounded by δ^ℓ.

Let us now consider the second event. Suppose message M was encrypted but message $M' \neq M$ gets decrypted in some slot $i \in [\ell]$, such a message passes all re-encryption checks and gets returned. Since $C_i = \mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M, i))$ but $\Pi.\mathsf{Dec}(\mathsf{sk}, C_i) = M'$, we deduce that the query $\mathsf{G}(M, i)$ is *problematic*. Moreover, since M' is returned, it means, that it passes all re-encryption checks, which in particular means that for all $j \in [\ell] \setminus \{i\}$

$$\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M, j)) = C_j = \mathsf{Enc}(\mathsf{pk}, M'; \mathsf{G}(M', j)).$$

Since $\Pi.\mathsf{Dec}$ is deterministic, at most one between M and M' can be equal to $\Pi.\mathsf{Dec}(\mathsf{sk}, C_j)$. Therefore, either $\mathsf{G}(M, j)$ or $\mathsf{G}(M', j)$ is *problematic*. As we have remarked before, G's outputs are independent, and each of them is *problematic* with probability at most δ. Thus, the overall probability of M' getting returned is bounded by δ^ℓ. Since there are in total $\ell - 1$ such possible indices i (recall that in the second event at least one ciphertext component correctly decrypts), a union bound shows that the probability of this second event happening is bounded by $(\ell - 1)\delta^\ell$.

$\Pi'.\mathsf{Enc}(\mathsf{pk}, M)$	$\Pi'.\mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$
for $i = 1, \ldots, \ell$ **do**	$\mathtt{res} \leftarrow \bot$, $\mathtt{check} \leftarrow \bot$
$\quad C_i := \Pi.\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M, i))$	**for** $i = 1, \ldots, \ell$ **do**
$C := (C_1, \ldots, C_\ell)$	$\quad \mathtt{res}[i] := \Pi.\mathsf{Dec}(\mathsf{sk}, C_i)$
return C	**for** $i \in [\ell]$ s.t. $\mathtt{res}[i] \neq \bot$ **do**
	$\quad$ **if** $\forall j \in [\ell] : \mathsf{ctxt}_j = \Pi.\mathsf{Enc}(\mathsf{pk}, \mathtt{res}[i]; \mathsf{G}(\mathtt{res}[i], j))$
	$\quad\quad \mathtt{check} \leftarrow i$
	if $\mathtt{check} \neq \bot$
	$\quad$ **return** $\mathtt{res}[\mathtt{check}]$
	return $\bot$

Figure 2.11: OW-PCA-secure scheme $\Pi' = \mathsf{T}^\star[\Pi, \mathsf{G}]$ with deterministic encryption and correctness error δ^ℓ from IND-CPA secure scheme Π with correctness error δ.

Putting everything together, we obtain that Π' has then correctness error $\delta' := \delta^\ell + (\ell - 1)\delta^\ell = \ell\delta^\ell$.

To the resulting PKE Π' we can then directly apply the transformation $\mathsf{U}^{\not\perp}$ to obtain an IND-CCA secure KEM KEM with negligible correctness error in the (Q)ROM.

Note that as we directly integrate the product compiler into the T transform, the correctness of the message can be checked via the de-randomization. Hence, we can get rid of the majority vote in the direct product compiler. With this change the analysis of the concrete choice of ℓ becomes simpler and, more importantly, allows us to choose smaller ℓ than in the black-box use of the compiler.

Remark 3. *Note that in Figure 2.11 we explicitly consider the case where* Dec *of the PKE scheme* Π *may return something arbitrary on failed decryption. For the simpler case where we have a PKE scheme* Π *which always returns* $\bot$ *on failed decryption, we can easily adapt the approach in Figure 2.11 and the correctness error analysis from above. Namely, we would decrypt all* ℓ *ciphertexts* ctxt_i, $i \in [\ell]$. *Let* $h \in [\ell]$ *be the minimum index such that* $\mathtt{res}[h] \neq \bot$. *Then for every element* $j \in [\ell]$ *run* $\mathsf{ctxt}'_j := \Pi.\mathsf{Enc}(\mathsf{pk}, \mathtt{res}[h]; \mathsf{G}(\mathtt{res}[h], j))$. *If for all* $j \in [\ell]$ *we have* $\mathsf{ctxt}'_j = \mathsf{ctxt}_j$ *we return* $\mathtt{res}[h]$. *If this is not the case we return* $\bot$. *Note that all* ℓ ctxt'_j *have to be computed and checked against* ctxt_j, *as otherwise* IND-CCA*-security is not achieved. The difference is, that only* ℓ *encryptions instead of* ℓ^2 *are required. As far as correctness error is concerned, it this case the correctness error is triggered if and only if no ciphertext correctly decrypts. This happens with probability* δ^ℓ.

We now show the following theorem.

Theorem 2.3.9 (Π IND-CPA $\implies$ Π' OW-PCA)**.** *Assume* Π *to be* ϵ*-key* δ*-correct. Then,* Π' *is* ϵ_1*-key* $\delta_1(q_\mathsf{G}, \ell)$*-correct in the random oracle model, for* $\epsilon_1 = \epsilon$ *and* $\delta_1(q_\mathsf{G}, \ell) \leq \frac{q_\mathsf{G}}{\ell} \cdot \ell \cdot \delta^\ell = q_\mathsf{G} \cdot \delta^\ell$. *For any* OW-PCA *adversary* B *that issues at most* q_G *queries to the random oracle* G *and* q_P *queries to a plaintext checking oracle* $P\textsc{co}$, *there exists an*

$\mathsf{IND\text{-}CPA}$ *adversary* A *running in about the same time as* B *such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi',B}(\lambda) \leq \epsilon + \left(\frac{q_\mathsf{G}}{\ell} + q_P\right) \cdot \ell\delta^\ell + \frac{2q_\mathsf{G}+1}{|\mathcal{M}|} + 3\ell \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,A}(\lambda).$$

We provide the proof which closely follows the proof of [HHK17b, Thm 3.2] in Appendix A.2.1. Note that we lose an additional factor of ℓ. Additionally, when using the bounded δ-correctness notion from Bindel. et al. [BS20], the factor of q_G disappears.

We now have an $\mathsf{OW\text{-}PCA}$ secure PKE Π' with negligible correctness error and can thus directly use $\mathsf{U}^{\not\perp}$ and by invoking Theorem 2.3.6 obtain an $\mathsf{IND\text{-}CCA}$ secure KEM KEM. Note that all steps in the reduction are tight. For the security in the QROM, we can directly conclude from Corollary 2.3.1 that the generic framework of Bindel et al. [BHH+19] can be applied to $\mathsf{C}_{\mathsf{p,d}}$ and $\mathsf{C}_{\mathsf{p,r}}$ with the additional constraint of ε-injectivity and FFC, respectively. Without these additional constraints, the results of Jiang et al. [JZM19] or Hövelmanns et al. [HKSU20][3] apply without the tighter reductions that the Bindel et al.'s and Kuchta et al.'s results offer.

The security of the $\mathsf{T}^\star$ transform in the QROM follows in a similar vein. To highlight how ℓ influences the advantages, we follow the proof strategy of Bindel et al. [BHH+19]. Therefore, we first show that a randomized $\mathsf{IND\text{-}CPA}$-secure PKE scheme with a non-negligible correctness error is transformed to $\mathsf{OW\text{-}CPA}$-secure deterministic PKE scheme with negligible correctness error. Second, we prove that if the $\mathsf{T}^\star$-transformed version is also ε-injective, then it provides FFC. With these two results in place, we can apply Theorem 2.3.8 to obtain an $\mathsf{IND\text{-}CCA}$-secure KEM.

In the following theorem, we prove $\mathsf{OW\text{-}CPA}$ security of the $\mathsf{T}^\star$ transform in the QROM (see Appendix A.1.1). We follow the strategy of the proof of [BHH+19, Thm. 1] and adapt it to our transform. Compared to the T transform, we lose a factor of ℓ^2. Once the loss is incurred by Theorem 2.2.3 and once by the semi-classical one-way to hiding Theorem [AHU19].

Theorem 2.3.10 (Π $\mathsf{IND\text{-}CPA}$ $\implies$ Π' $\mathsf{OW\text{-}CPA}$)**.** *Let* Π *be a non-deterministic PKE with randomness space* $\mathcal{R}$ *and decryption error* δ*. Let* $\ell \in \mathbb{N}$ *such that* δ^ℓ *is negligible in the security parameter* λ*. Let* $\mathsf{G}\colon \mathcal{M} \times [\ell] \to \mathcal{R}$ *be a quantum-accessible random oracle and let* q_G *the number queries with depth at most* d*. If* $\mathcal{A}$ *is an* $\mathsf{OW\text{-}CPA}$*-adversary against* $\mathsf{T}^\star[\Pi, \mathsf{G}, \ell]$*, then there exists an* $\mathsf{IND\text{-}CPA}$ *adversary* B *against* Π*, running in about same time as* $\mathcal{A}$*, such that*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\mathsf{T}^\star[\Pi,\mathsf{G},\ell],\mathcal{A}}(\lambda) \leq (d+\ell+1) \cdot \left(\ell \cdot \mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,B}(\lambda) + \frac{8 \cdot (q_\mathsf{G}+1)}{|\mathcal{M}|}\right).$$

We refer to Appendix A.2.2 for the proof. Next, we show that the transform provides the FFC property (cf. [BHH+19, Lemma 6]).

[3] Without restating [HKSU20, Thm 3.2], note that we can adopt it the same way we highlight in Theorems 2.3.9 and 2.3.10. So, we start with their Punc to obtain disjoint simutability and then apply $\mathsf{T}^\star$ and $\mathsf{U}^{\not\perp}_m$.

Table 2.2: Comparison of the runtime and bandwidth overheads of $\mathsf{C}_{\mathsf{p},\mathsf{y}}$, $\mathsf{y} \in \{\mathsf{r}, \mathsf{d}\}$, with ℓ ciphertexts and $\mathsf{T}^\star$ and $\mathsf{C}_{\mathsf{p},\mathsf{d}}{}^\star$ with ℓ' ciphertexts such that $\ell \geq \ell' + 1$.

	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{ctxt}\rvert$	KGen	Enc	Dec
$\mathsf{C}_{\mathsf{p},\mathsf{y}}$	1 (r) / ℓ (d)	ℓ	1 (r) / ℓ (d)	ℓ	ℓ
$\mathsf{C}_{\mathsf{p},\mathsf{d}}{}^\star$	ℓ'	ℓ'	ℓ'	ℓ'	ℓ'
$\mathsf{T}^\star$	1	ℓ'	1	ℓ'	ℓ'^2 / ℓ' ($\perp$)

Lemma 2.3.1. *If Π is a δ-correct non-deterministic PKE with randomness space $\mathcal{R}$, $\ell \in \mathbb{N}$ such that δ^ℓ is negligible in the security parameter λ, $\mathsf{G}\colon \mathcal{M} \times [\ell] \to \mathcal{R}$ is a random oracle so that $\Pi' = \mathsf{T}^\star[\Pi, \mathsf{G}, \ell]$ is ε-injective, then the advantage for any FFC-adversary $\mathcal{A}$ against Π' which makes at most q_G queries at depth d to G and which returns a list of at most q_L ciphertexts is bounded by*

$$\mathsf{Adv}^{\mathsf{pke\text{-}ffc}}_{\Pi',\mathcal{A}}(\lambda) \leq \left((4 \cdot d + 1) \cdot \delta^\ell + \sqrt{3 \cdot \varepsilon}\right) \cdot (q_\mathsf{G} + q_L) + \varepsilon.$$

For the proof we refer to Appendix A.2.3.

2.3.5 Comparison of the Two Approaches

The major difference between the generic approach using the direct product compiler $\mathsf{C}_{\mathsf{p},\mathsf{y}}$, $\mathsf{y} \in \{\mathsf{r}, \mathsf{d}\}$, and $\mathsf{T}^\star$ (or the modified deterministic direct product compiler $\mathsf{C}_{\mathsf{p},\mathsf{d}}{}^\star$) is the number of ciphertexts required to reach a negligible correctness error. As observed in Section 2.3.2, the analysis of the overall decryption error is rather complicated and $\mathsf{C}_{\mathsf{p},\mathsf{y}}$ requires at least $\ell \geq 3$. With $\mathsf{T}^\star/\mathsf{C}_{\mathsf{p},\mathsf{d}}{}^\star$ however, the situation is simpler. As soon as one ciphertext decrypts correctly and no encryption collision happen, the overall correctness of the decryption can be guaranteed. Also, for the cases analysed in Table 2.1, $\mathsf{C}_{\mathsf{p},\mathsf{y}}$ requires at least one ciphertext more than $\mathsf{T}^\star$ and $\mathsf{C}_{\mathsf{p},\mathsf{d}}{}^\star$. For the correctness error, we have a loss in the number of random oracle queries in both cases. For the comparison of the runtime and bandwidth overheads, we refer to Table 2.2. Note that if the Dec of the underlying PKE Π reports decryption failures with $\perp$, then the overhead of $\mathsf{T}^\star$ for Dec is only a factor ℓ (cf. Remark 3).

2.4 Our Transform in Practice

The most obvious use-case for IND-CCA secure KEMs in practice is when considering static long-term keys. Systems supporting such a setting are for example RSA-based key exchange for SSH [Har06] or similarly in TLS up to version 1.2. But since the use of long-term keys precludes forward-secrecy guarantees, using static keys is not desirable. For ephemeral keys such as used in the ephemeral Diffie-Hellman key exchange, an IND-CPA secure KEM might seem sufficient. Yet, in the post-quantum setting accidental re-use of an ephemeral key leads to a wide range of attacks [BGRR19]. But also from

a theoretical viewpoint it is unclear whether CPA security actually would be enough. Security analysis of the TLS handshake protocol suggests that in the case of version 1.2 an only passively secure version is insufficient [JKSS12, KPW13] (cf. also [PST20]). Also, security analysis of the version 1.3 handshake requires IND-CCA security [DFGS15]. Thus, even in the case of ephemeral key exchanges, using a IND-CCA secure KEM is actually desirable and often even necessary as highlighted by Schwabe et al. [SSW20b].

For comparing KEMs in this context, the interesting metric is hence not the ciphertext size alone, but the combined public key and ciphertext size. Both parts influence the communication cost of the protocols. Additionally, the combined runtime of the key generation, encapsulation and decapsulation is also an interesting metric. All three operations are performed in a typical ephemeral key exchange and hence give a lower bound for the overall runtime of the protocol.

In the following comparison, we assume that the underlying PKE never returns $\perp$ on failure, but an incorrect message instead. Thereby we obtain an upper bound for the runtime of the Decaps algorithm. For specific cases where Decaps explicitly returns $\perp$ on failure, the runtime figures would get better since the overhead to check the ciphertexts is reduced to a factor of ℓ (cf. Remark 3).

2.4.1 Code-Based KEMs

KEMs based on error correcting codes can be parametrized such that the decoding failure rate (DFR) is non-negligible, negligible, or 0. Interestingly, the DFR rate is also influenced by the actual decoder. Even for the same choice of code and the exact same instance of the code, a decoder might have a non-negligible DFR, whereas another (usually more complex) decoder obtains a negligible DFR. For the submissions in the NIST PQC we can observe all three choices. The candidates providing IND-CPA-secure variants with non-negligible DFR include: BIKE [ABB+19], ROLLO [ABD+19], and LEDAcrypt [BBC+19]. We discuss the application of our transform to those schemes below. For the comparison in Table 2.3, we consider the DFR as upper bound for correctness error. In Table 2.3, we present an overview of the comparison (see Appendix A.3 for the full comparison). First we consider ROLLO, and in particular ROLLO-I, where we obtain the best results: public key and ciphertext size combined is always smaller than for ROLLO-II and the parallel implementation is faster even in case of a ℓ^2 overhead. For both BIKE (using $\mathsf{T}^\star$) and LEDAcrypt (using $\mathsf{C}^\star_{\mathsf{p,d}}$ since it starts from a deterministic PKE), we observe a trade-off between bandwidth and runtime.

2.4.2 Lattice-Based KEMs

For lattice-based primitives the decryption error depends both on the modulus q and the error distribution used.

As discussed in [SAB+19], an important decision that designers have to make is whether to allow decryption failures or choose parameters that not only have a negligible, but a zero chance of failure. Having a perfectly correct encryption makes transforms to

Table 2.3: Sizes (in bytes) and runtimes (in ms and millions of cycles for BIKE), where O denotes the transformed scheme. The LEDAcrypt instances with postfix NN refer to those with non-negligible DFR. Runtimes are taken from the respective submission documents and are only intra-scheme comparable.

KEM	δ	pk	ctxt	$\sum$	KGen	Encaps	Decaps
O[ROLLO-I-L1,5]	$2^{-147.7}$	**465**	2325	**2790**	**0.10**	**0.02**/0.10	**0.26**/1.30
ROLLO-II-L1	2^{-128}	1546	1674	3220	0.69	0.08	0.53
O[ROLLO-I-L3,4]	2^{-126}	**590**	2360	**2950**	**0.13**	**0.02**/**0.08**	**0.42**/1.68
ROLLO-II-L3	2^{-128}	2020	2148	4168	0.83	0.09	0.69
O[ROLLO-I-L5,4]	2^{-166}	**947**	7576	8523	**0.20**	**0.03**/0.12	**0.78**/3.12
ROLLO-II-L5	2^{-128}	2493	2621	5114	0.79	0.10	0.84
O[BIKE-2-L1,3]	$2^{-145.4}$	**10163**	30489	40652	**4.79**	**0.14**/0.42	**3.29**/9.88
BIKE-2-CCA-L1	2^{-128}	11779	12035	23814	6.32	0.20	4.12
O[LEDAcrypt-L5-NN,2]	2^{-127}	22272	22272	44544	5.04	**0.14**/**0.29**	**1.55**/3.11
LEDAcrypt-L5	2^{-128}	19040	19040	38080	4.25	0.84	2.28

obtain IND-CCA security and security proofs easier, but with the disadvantage that this means either decreasing security against attacks targeting the underlying lattice problem or decreasing performance. The only NIST PQC submissions based on lattices which provide parameter sets achieving both negligible and non-negligible decryption failure are ThreeBears [Ham19] and Round5 [GZB+19]. The IND-CCA-secure version of ThreeBears is obtained by tweaking the error distribution, hence, our approach does not yield any improvements. For Round5 we achieve a trade-off between bandwidth and runtime. We also considered FrodoKEM [NAB+19], comparing its version [BCD+16] precedent to the NIST PQC, which only achieved non-negligible failure probability, to the ones in the second round of the above competition, but we do not observe any improvements for this scheme. For the full comparison we refer to Appendix A.3. It would be interesting to understand the reasons why the compiler does not perform well on lattice-based scheme compared to the code-based ones and whether this is due to the particular schemes analysed or due to some intrinsic difference between code- and lattice-based constructions.

2.4.3 Implementation Aspects

One of the strengths of $\mathsf{T}^\star$ compared to the black-box use of $\mathsf{C}_{\mathsf{p},\mathsf{y}}$, $\mathsf{y} \in \{\mathsf{r}, \mathsf{d}\}$ (and $\mathsf{C}_{\mathsf{p},\mathsf{d}}{}^\star$), is that besides the initial generation of the encapsulated key, all the random oracle calls can be evaluated independently. Therefore, the encryptions of the underlying PKE do not depend on each other. Thus, the encapsulation algorithms are easily parallelizable – both in software and hardware. The same applies to the decapsulation algorithm. While in this case only one successful run of the algorithm is required, doing all of them in parallel helps to obtain a constant-time implementation. Then, after all ciphertexts have

been processed, the first valid one can be used to re-compute the ciphertexts, which can be done again in parallel. For software implementations on multi-core CPUs as seen on today's desktops, servers, and smartphones with 4 or more cores, the overhead compared to the IND-CPA secure version is thus insignificant as long as the error is below 2^{-32}. If not implemented in a parallel fashion, providing a constant-time implementation of the decapsulation algorithms is more costly. In that case, all of the ciphertexts have to be dealt with to not leak the index of invalid ciphertexts. Note that a constant-time implementation of the transform is important to avoid key-recovery attacks [GJN20].

The $\mathsf{T}^\star$ transform also avoids new attack vectors such as [GJY19] that are introduced via different techniques to decrease the correctness error, e.g., by applying an error-correcting code on top. Furthermore, since the same parameter sets are used for the IND-CPA and IND-CCA secure version when applying our transforms, the implementations of proposals with different parameter sets can be simplified. Thus, more focus can be put on analysing one of the parameter sets and also on optimizing the implementation of one of them.

2.5 Application to Bloom Filter KEMs

A Bloom Filter Key Encapsulation Mechanism (BFKEM) [DJSS18, DGJ+21] is a specific type of a puncturable encryption scheme [GM15, GHJL17, DJSS18, SSS+20] where one associates a Bloom Filter (BF) [Blo70] to its public-secret key pair depending on the BF-parameters $k, m \in \mathbb{N}$. The initial (i.e., non-punctured) secret key is associated to an empty BF where all bits are set to 0. (In particular, the BF allows for a compact binary representation T of $[m]$.) Encapsulation, depending on a so-called tag u in the universe of the BF, takes the public key, and returns a ciphertext and an encapsulation key k corresponding to the BF-evaluation of u, i.e., k hash evaluations on u yielding so-called indexes in the domain $[m]$. Puncturing, on input a ciphertext ctxt (associated to tag u) and a secret key sk', punctures sk' on ctxt and returns the resulting secret key. Decapsulation, on input a ciphertext ctxt (with an associated tag u) and secret key sk' is able to decapsulate the ciphertext to k if sk' was not punctured on ctxt. We want to mention, as in [DGJ+21], we solely focus on KEMs since a Bloom Filter Encryption (BFE) scheme (which encrypts a message from some message space) can be generically derived from a BFKEM (cf. [FO99]).

The basic instantiation of a BFKEM in [DJSS18, DGJ+21] is non-black box and based on the pairing-based Boneh-Franklin Identity-Based Encryption (IBE) scheme [BF01], where sk contains an IBE secret key for every "identity" $i \in [m]$ of the BF bits (according to T) and puncturing amounts to inserting tag u in the BF and deleting the IBE secret keys for the corresponding bits. Although the BFKEM is defined with respect to a non-negligible correctness error, the underlying variant of the Boneh-Franklin IBE has perfect correctness. So the non-negligible error in the BFKEM is only introduced on an abstraction (at the level of the BF) above the Fujisaki-Okamoto (FO) transform [FO99, FO13] applied to the k Boneh-Franklin IBE ciphertexts (so the application of the FO can be done as usual for perfectly correct encryption schemes).

However, if one targets instantiations of BFKEM where the underlying IBE does not have perfect correctness (e.g., lattice- or code-based IBEs), it is not obvious whether the security proof using the Boneh-Franklin IBE as presented in [DJSS18, DGJ+21] can easily be adapted to this setting.[4]

We first recall necessary definitions for BFs, BFKEMS, and their properties from [DGJ+21] and show a generic construction of BFKEM from any IBE scheme with (non-)negligible correctness error in Section 2.5.1.

Definition 2.5.1 (Bloom Filter)**.** *A Bloom Filter (BF) [Blo70]* BF *consists of the PPT algorithms* $(\mathsf{BF\text{-}Gen}, \mathsf{BF\text{-}Update}, \mathsf{BF\text{-}Check})$*:*

$\mathsf{BF\text{-}Gen}(m,k)$: *BF generation, on input BF parameters* $m,k \in \mathbb{N}$*, samples* k *universal hash functions* $H_1,\dots,H_k$*, where* $H_j\colon \mathcal{U} \to [m]$*, for all* $j \in [k]$*, defines* $H := (H_j)_{j\in[k]}$*, sets* $T_0 := 0^m$*, i.e., an* m*-bit array of all* 0*, and outputs* (H, T_0)*.*

$\mathsf{BF\text{-}Update}(H,T,u)$: *The BF-update algorithm, on input* $H = (H_j)_{j\in[k]}$*,* $T \in \{0,1\}^m$*, and* $u \in \mathcal{U}$*, sets* $T' := T$ *and, afterwards,* $T'[H_j(u)] := 1$*, where* $T'[i]$ *denotes the* i*-th bit of* T'*, for all* $j \in [k]$*. The algorithms outputs the updated state* T'*.*

$\mathsf{BF\text{-}Check}(H,T,u)$: *The BF-check algorithm, on input* $H = (H_j)_{j\in[k]}$*,* $T \in \{0,1\}^m$*, and* $u \in \mathcal{U}$*, returns a bit* $b := \bigwedge_{j\in[k]} T[H_j(u)]$*, where* $T[i]$ *denotes the* i*-th bit of* T*.*

For all $m,k \in \mathbb{N}$*, we require the following properties of* BF*:*

Perfect completeness. *For all* $(H,T_0) \leftarrow \mathsf{BF\text{-}Gen}(m,k)$*, for all* $n \in \mathbb{N}$*, for all* $(u_1,\dots,u_n) \in \mathcal{U}^n$*, for all* $i \in [n]$*, for all* $T_i \leftarrow \mathsf{BF\text{-}Update}(H,T_{i-1},u_i)$*, we require that* $\mathsf{BF\text{-}Check}(H,T_n,u_i) = 1$ *holds.*

Compact representation of any $\mathcal{U}' \subset \mathcal{U}$**.** *The size of the any representation* T_i*, for all* T_i *as output of* $\mathsf{BF\text{-}Update}$*, is a constant number of* m *bits independent of the size of any set* $\mathcal{U}' \subset \mathcal{U}$ *and the representation of any element in* $\mathcal{U}$*.*

Bounded false-positive probability. *For all* $(H,T_0) \leftarrow \mathsf{BF\text{-}Gen}(m,k)$*, for all* $n \in \mathbb{N}$*, for all* $\mathcal{U}' = (u_1,\dots,u_n) \in \mathcal{U}^n$*, for all* $i \in [n]$*, for all* $T_i \leftarrow \mathsf{BF\text{-}Update}(H,T_{i-1},u_i)$*, for all* $u^* \in \mathcal{U}\backslash\mathcal{U}'$*, we require that* $\Pr[\mathsf{BF\text{-}Check}(H,T_n,u^*) = 1] \leq \left(1 - e^{-\frac{(n+1/2)k}{m-1}}\right)^k$ *holds, where the probability is taken over the random coins of* $\mathsf{BF\text{-}Gen}$*.*

In the following, we recap the BFKEM and its formal properties from [DGJ+21] which tolerates a non-negligible correctness error and the key generation takes parameters m and k

[4] For practical reasons, we want the size of the BFKEM public key to be independent of the BF parameters (besides the descriptions of the hash functions). Right now, we only can guarantee this with IBE schemes as such schemes allow for exponentially many identity-based secret keys (in the security parameter) while maintaining a short public key.

as input which specify the correctness error. Furthermore, we slightly adapt their BFKEM properties *extended correctness*, *separable randomness*, and *publicly-checkable puncturing* to allow a negligible decryption error for extended correctness and publicly-checkable puncturing properties while extending the input space for the separable randomness property.

Definition 2.5.2 (Bloom Filter Key Encapsulation Mechanism)**.** *A BFKEM* BFKEM *with key space* $\mathcal{K}$ *consists of the PPT algorithms* (KGen, Encaps, Punc, Decaps)*.*

$\mathsf{KGen}(\lambda, m, k)$: *Key generation, on input security parameter* λ *and BF parameters* m, k*, outputs public and secret keys* $(\mathsf{pk}, \mathsf{sk}_0)$*. (We assume that* pk *is available to* Punc *and* Decaps *implicitly.)*

$\mathsf{Encaps}(\mathsf{pk})$: *Encapsulation, on input* pk*, outputs a ciphertext* ctxt *and key* k*.*

$\mathsf{Punc}(\mathsf{sk}, \mathsf{ctxt})$: *Secret-key puncturing, on input* sk *and* ctxt*, outputs an updated secret key* sk'*.*

$\mathsf{Decaps}(\mathsf{sk}, \mathsf{ctxt})$: *Decapsulation, on input* sk *and* ctxt*, outputs* k *or* $\{\perp\}$*.*

Definition 2.5.3 (Correctness of BFKEM)**.** *For all* $\lambda, m, k, n \in \mathbb{N}$ *and any* $(\mathsf{pk}, \mathsf{sk}_0) \leftarrow \mathsf{KGen}(\lambda, m, k)$*, we require that for any (arbitrary interleaved) sequence of invocations of* $\mathsf{sk}_i \leftarrow \mathsf{Punc}(\mathsf{sk}_{i-1}, \mathsf{ctxt}_{i-1})$*, for* $(\mathsf{ctxt}_{i-1}, \mathsf{k}_{i-1}) \leftarrow \mathsf{Encaps}(\mathsf{pk})$*, for* $i \in [n]$*, it holds that*

$$\Pr\left[\mathsf{Decaps}(\mathsf{sk}_n, \mathsf{ctxt}_n) \neq \mathsf{k}_n\right] \leq \left(1 - e^{-\frac{(n+1/2)\cdot k}{m-1}}\right)^k + \varepsilon(\lambda),$$

where $(\mathsf{ctxt}_n, \mathsf{k}_n) \leftarrow \mathsf{Encaps}(\mathsf{pk})$ *and* ε *is a negligible function in* λ*. The probability is taken over the random coins of* KGen*,* Encaps*, and* Punc*.*

Definition 2.5.4 (Extended Correctness of BFKEM)**.** *For all* $\lambda, m, k, n \in \mathbb{N}$ *and any* $(\mathsf{pk}, \mathsf{sk}_0) \leftarrow \mathsf{KGen}(\lambda, m, k)$*, we require that for any (arbitrary interleaved) sequence of invocations of* $\mathsf{sk}_i \leftarrow \mathsf{Punc}(\mathsf{sk}_{i-1}, C_{i-1})$*, where* $i \in [n]$ *and* $(C_{i-1}, \mathsf{k}_{i-1}) \leftarrow \mathsf{Encaps}(\mathsf{pk})$*, it holds that:*

(a) Impossibility of false-negatives: $\mathsf{Decaps}(\mathsf{sk}_n, C_{j-1}) = \perp$*, for all* $j \in [n]$*.*

(b) Correctness of the initial secret key: $\Pr[\mathsf{Decaps}(\mathsf{sk}_0, C) \neq \mathsf{k}] \leq \varepsilon(\lambda)$*, for all* $(C, \mathsf{k}) \leftarrow \mathsf{Encaps}(\mathsf{pk})$ *and* ε *is a negligible function in* λ*.*

(c) Semi-correctness of punctured secret keys: if $\mathsf{Decaps}(\mathsf{sk}_j, C) \neq \perp$ *then* $\Pr[\mathsf{Decaps}(\mathsf{sk}_j, C) \neq \mathsf{Decaps}(\mathsf{sk}_0, C)] \leq \varepsilon(\lambda)$*, for all* $j \in [n]$*, any* C*, and* ε *is a negligible function in* λ*.*

All probabilities are taken over the random coins of KGen, Punc, *and* Encaps*. The difference to [DGJ+21] is that we allow for a negligible error in (b) and (c).*

Definition 2.5.5 (Separable Randomness of BFKEM)**.** *For all $\lambda, m, k \in \mathbb{N}$, for $(\mathsf{pk}, \cdot) \leftarrow \mathsf{KGen}(\lambda, m, k)$, a BFKEM* BFKEM *has the property* separable randomness *if the encapsulation algorithm* Encaps *can be written as*

$$(\mathsf{ctxt}, \mathsf{k}) \leftarrow \mathsf{Encaps}(\mathsf{pk}) = \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k})),$$

for some $(r, \mathsf{k}) \in \mathcal{R} \times \mathcal{K}$, for randomness space $\mathcal{R} = \underbrace{\{0,1\}^\rho \times \cdots \times \{0,1\}^\rho}_{k \text{ times}}$ and key space $\mathcal{K}$ of BFKEM*, for large-enough integer ρ. Hence, pk, r and k as input to deterministic* Encaps *uniquely determine $(\mathsf{ctxt}, \mathsf{k})$. The difference to [DGJ+21] is that we extend the randomness space as input to* Encaps*.*

Definition 2.5.6 (Publicly-Checkable Puncturing of BFKEM)**.** *For all $\lambda, m, k, \ell \in \mathbb{N}$,* BFKEM *has the* publicly-checkable puncturing *property if there exists a PPT algorithm* CheckPunct *such that after running $(\mathsf{pk}, \mathsf{sk}_0) \leftarrow \mathsf{KGen}(\lambda, m, k)$, $(\mathsf{ctxt}_{i-1}, \mathsf{k}_{i-1}) \leftarrow \mathsf{Encaps}(\mathsf{pk})$, and $\mathsf{sk}_i \leftarrow \mathsf{Punc}(\mathsf{sk}_{i-1}, \mathsf{ctxt}_{i-1})$, for $i \in [\ell]$, we have that*

$$\Pr\left[\mathsf{Decaps}(\mathsf{sk}_\ell, C) = \bot \iff \mathsf{CheckPunct}(\mathsf{pk}, \mathcal{L}, C) = \bot\right] \leq \varepsilon(\lambda),$$

holds, for $\mathcal{L} = (\mathsf{ctxt}_0, \ldots, \mathsf{ctxt}_{\ell-1})$, for any C, and ε is a negligible function in λ. The probability is taken over the random coins of KGen, Punc, *and* Encaps*.*

Definition 2.5.7 (γ-Spreadness of BFKEM)**.** *For all $\lambda, m, k, \rho \in \mathbb{N}$, a BFKEM* BFKEM *with separable randomness is γ-*spread*, if for any $(\mathsf{pk}, \cdot) \leftarrow \mathsf{KGen}(\lambda, m, k)$, any keys $\mathsf{k} \in \mathcal{K}$, $r \leftarrow \mathcal{R}$, and any $C \in \mathcal{C}$, where $\mathcal{R}$ and $\mathcal{C}$ are the randomness and ciphertext spaces of* BFKEM*, respectively, we have that $\Pr[(C, \cdot) = \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k}))] \leq 2^{-\gamma}$ holds, where the probability is taken over the random coins of* KGen*.*

BFKEM-IND-CPA and BFKEM-IND-CCA security. We say a BFKEM BFKEM is BFKEM-IND-CPA or BFKEM-IND-CCA secure if and only if any PPT adversary $\mathcal{A}$ has only negligible advantage in the following security experiments. First, $\mathcal{A}$ gets an honestly generated public key pk as well as a ciphertext-key pair $(\mathsf{ctxt}^*, \mathsf{k}_b^*)$, for $(\mathsf{ctxt}^*, \mathsf{k}_0) \leftarrow \mathsf{Encaps}(\mathsf{pk})$, for $\mathsf{k}_1 \leftarrow \mathcal{K}$, and for $b \leftarrow \{0,1\}$. Furthermore, $\mathcal{A}$ has access to Punc'-, Cor-, and $\underline{\mathsf{Decaps}'}$-oracle (with initially empty set $\mathcal{L}$ with $\ell := 0$ and the latter oracle only in the BFKEM-IND-CCA-security experiment):

$\mathsf{Punc}'(\mathsf{ctxt})$: on input ctxt, set $\mathcal{L} := \mathcal{L} \cup \{\mathsf{ctxt}\}$ and $\ell := \ell + 1$, compute $\mathsf{sk}_\ell \leftarrow \mathsf{Punc}(\mathsf{sk}_{\ell-1}, \mathsf{ctxt})$, store and return sk_ℓ.

Cor: if $\mathsf{ctxt}^* \in \mathcal{L}$, then return sk_ℓ, else outputs $\bot$.

$\underline{\mathsf{Decaps}'(\mathsf{ctxt})}$: on input ctxt, if $\mathsf{ctxt} \neq \mathsf{ctxt}^*$, then return $\mathsf{Decaps}(\mathsf{sk}_0, \mathsf{ctxt})$, else return $\bot$.

Eventually, $\mathcal{A}$ outputs a guess b'. Finally, if $b = b'$, then the experiment outputs 1. The formal experiments are depicted in Figure 2.12.

Exp. $\mathsf{Exp}^{\mathsf{bfkem\text{-}ind\text{-}y}}_{\mathsf{BFKEM},\mathcal{A}}(\lambda, m, k)$
$(\mathsf{pk}, \mathsf{sk}_0) \leftarrow \mathsf{KGen}(\lambda, m, k)$
$(\mathsf{ctxt}^*, \mathsf{k}_0^*) \leftarrow \mathsf{Encaps}(\mathsf{pk}), \mathsf{k}_1^* \leftarrow \mathcal{K}$
$b \leftarrow \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathsf{Punc}'(\cdot),\mathsf{Cor},\underline{\mathsf{Decaps}'(\cdot)}}(\mathsf{pk}, \mathsf{ctxt}^*, \mathsf{k}_b^*)$
if $b = b'$ then return 1 else return 0

Figure 2.12: BFKEM-IND-y security experiments for BFKEM, for $\mathsf{y} \in \{\mathrm{CPA}, \mathrm{CCA}\}$. The differences between BFKEM-IND-CPA and BFKEM-IND-CCA are given by underlining.

$\underline{\mathsf{Enc}'(\mathsf{mpk}, id, M)}$	$\underline{\mathsf{Dec}'(\mathsf{usk}[id], \mathsf{ctxt})}$
for $i \in [\ell]$	$\mathsf{ctxt} =: (\mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell)$
$r_i \leftarrow \mathcal{R}$	**for** $i \in [\ell]$
$\mathsf{ctxt}_i \leftarrow \mathsf{Enc}(\mathsf{mpk}, id, M; r_i)$	$M'_i := \mathsf{Dec}(\mathsf{usk}[id], \mathsf{ctxt}_i)$
return $(\mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell)$	**return** $\mathsf{maj}(M'_1, \ldots, M'_\ell)$

Figure 2.13: Compiler for Enc' and Dec' for constructing an IBE scheme IBE' with negligible correctness error from an IBE scheme IBE with non-negligible correctness error.

Definition 2.5.8. *For any PPT adversary $\mathcal{A}$ and all $\lambda, m, k \in \mathbb{N}$, the advantage functions*

$$\mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}y}}_{\mathsf{BFKEM},\mathcal{A}}(\lambda, m, k) := \left| \Pr[\mathsf{Exp}^{\mathsf{bfkem\text{-}ind\text{-}y}}_{\mathsf{BFKEM},\mathcal{A}}(\lambda, m, k) = 1] - \frac{1}{2} \right|,$$

for $\mathsf{y} \in \{\mathsf{cpa}, \mathsf{cca}\}$, are negligible in λ, where the experiments $\mathsf{Exp}^{\mathsf{bfkem\text{-}ind\text{-}y}}_{\mathsf{BFKEM},\mathcal{A}}(\lambda, m, k)$ are given in Figure 2.12 and BFKEM is a BFKEM.

2.5.1 IBE with Negligible from Non-Negligible Correctness Error

We follow the approach for randomized PKE schemes in Section 2.3.2 adapted for the IBE case (cf. Figure 2.13).[5] Let $\mathsf{IBE} = (\mathsf{KGen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE scheme with identity, message spaces, and randomness spaces $\mathcal{ID}$, $\mathcal{M}$, and $\mathcal{R}$, respectively, that is ϵ-key δ-correct for some *non-negligible correctness error* $\delta(\lambda)$, we construct an IBE scheme $\mathsf{IBE}' = (\mathsf{KGen}', \mathsf{Ext}', \mathsf{Enc}', \mathsf{Dec}')$ with identity and message spaces $\mathcal{ID}' := \mathcal{ID}$ and $\mathcal{M}' := \mathcal{M}$, respectively, that is ϵ-key δ'-correct, with *negligible correctness error* $\delta'(\lambda)$. The construction is as follows. Set $\mathsf{KGen}' := \mathsf{KGen}$ and $\mathsf{Ext}' := \mathsf{Ext}$ while Enc' and Dec' are given in Figure 2.13. See that $\ell = \ell(\lambda)$ can be chosen appropriately to accommodate a negligible correctness error $\delta'(\lambda)$.

[5] We explicitly mention that we are only concerned with randomized IBEs. Adopting $\mathsf{C}_{\mathsf{p,d}}$ for deterministic IBEs will work as well. Though in the latter case, one can further optimize the compiler depending on whether the IBE has deterministic or randomized key extraction Ext.

As for randomized PKE schemes, by an analogue of Theorem 2.2.3 for IBEs with $q = \ell$ and $n = 1$, the security claim follows:

Corollary 2.5.1. *For any* IBE-sIND-CPA *adversary* B *against* IBE′ *obtained via applying the above transformation to* IBE*, there exists an* IBE-sIND-CPA *adversary* A *such that*

$$\mathsf{Adv}^{\text{ibe-sind-cpa}}_{\mathsf{IBE}',B}(\lambda) \leq \ell \cdot \mathsf{Adv}^{\text{ibe-sind-cpa}}_{\mathsf{IBE},A}(\lambda).$$

The correctness-error analysis is again equivalent to the one in the PKE scenario. We refer to Section 2.3.2 for a more in-depth discussion.

2.5.2 BFKEM from IBE with Negligible Correctness Error

The intuition for our generic construction from any IBE scheme IBE with negligible correctness error is as follows. We associate "user-secret keys" of IBE with the indexes $i \in [m]$ of the Bloom filter BF and annotate sk'_0 as a special key for "fixed identity" 0. We consider the encapsulation key as $\mathsf{k} = (\mathsf{k}_0, \mathsf{k}_1)$ where the first share is encrypted under "identity" 0 (yielding ctxt'_0) while the other share is encrypted under the "identities" $(i_j)_j$ of indexes of the BF that are determined by ctxt'_0. Put differently, ctxt'_0 acts as a tag of the overall ciphertext while the other IBE-ciphertexts $(\mathsf{ctxt}'_{i_j})_j$ are utilized for correct decryption, i.e., the secret key is punctured on "tag" ctxt'_0. Note that the secret key sk'_0 is not affected by the puncturing mechanism and one can always at least decrypt ctxt'_0. However, one additionally needs the encapsulation-key share from the other IBE-ciphertexts $(\mathsf{ctxt}'_{i_j})_j$; those ciphertexts can only be decrypted if at least one secret key $\mathsf{sk}'_{i'}$, for some index $i' \in [m]$, is available which can be checked with BF-Check.

More concretely, let $\mathsf{BF} = (\mathsf{BF\text{-}Gen}, \mathsf{BF\text{-}Update}, \mathsf{BF\text{-}Check})$ be a BF with universe $\mathcal{U}$ and BF (integer) parameters $m, k \in \mathbb{N}$. Furthermore, let $\mathsf{IBE} = (\mathsf{IBE.KGen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ be an IBE-sIND-CPA-secure IBE scheme with identity space $[m] \cup \{0\}$, message space $\mathcal{M}$, and negligible key and correctness error $\epsilon = \epsilon(\lambda)$ and $\delta = \delta(\lambda)$. We construct a BFKEM-IND-CPA-secure BFKEM scheme $\mathsf{BFKEM} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Punc}, \mathsf{Decaps})$ with key space $\mathcal{K} := \mathcal{M} \times \mathcal{M}$ and non-negligible correctness error $\delta' = \delta'(\lambda, m, k, n)$ in Figure 2.14. Later, we show how to use the BFKEM-IND-CPA-secure BFKEM with additional BFKEM properties (i.e., extended correctness, separable randomness, publicly-checkable puncturing, and γ-spreadness) as a stepping stone to build a BFKEM-IND-CCA-secure BFKEM.

Correctness of BFKEM**.** According to Definition 2.5.3, we have to show

$$\Pr[\mathsf{Decaps}(\mathsf{sk}_n, \mathsf{ctxt}_n) \neq \mathsf{k}_n] \leq (1 - e^{-\frac{(n+1/2)\cdot k}{m-1}})^k + \varepsilon(\lambda). \tag{2.1}$$

We argue that this holds due to the bounded false-positive probability of BF and due to the negligible IBE key and correctness error terms $\epsilon = \epsilon(\lambda)$ and $\delta = \delta(\lambda)$, with $\epsilon(\lambda) + \delta(\lambda) \leq \varepsilon(\lambda)$ for some negligible function $\varepsilon(\lambda)$ and for any number of punctures n. Concretely, see that Punc deletes IBE secret keys depending on the BF evaluated on

$\underline{\mathsf{KGen}(\lambda, m, k)}$:
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{IBE.KGen}(\lambda)$
$(H, T_0) \leftarrow \mathsf{BF\text{-}Gen}(m, k)$
$\mathsf{sk}'_{id} \leftarrow \mathsf{Ext}(\mathsf{msk}, id),\ id \in [m] \cup \{0\}$
$\mathsf{pk} := (\mathsf{mpk}, H), \mathsf{sk} := (T_0, (\mathsf{sk}'_{id})_{id})$
return $(\mathsf{pk}, \mathsf{sk}_0)$

$\underline{\mathsf{Punc}(\mathsf{sk}_{i-1}, \mathsf{ctxt})}$:
$(T, \mathsf{sk}'_0, (\mathsf{sk}'_{id})_{id\in[m]}) := \mathsf{sk}_{i-1}$
$(\mathsf{ctxt}_0, \ldots) := \mathsf{ctxt}$
$T' := \mathsf{BF\text{-}Update}(H, T, \mathsf{ctxt}_0)$
$\mathsf{sk}''_{id} := \begin{cases} \mathsf{sk}'_{id} & \text{if } T'[id] = 0, \\ \bot & \text{if } T'[id] = 1, \end{cases}$
return $(T', \mathsf{sk}'_0, (\mathsf{sk}''_{id})_{id\in[m]})$

$\underline{\mathsf{Encaps}(\mathsf{pk})}$:
$(\mathsf{mpk}, H) := \mathsf{pk}$ with $(H_j)_{j\in[k]} := H$
$(\mathsf{k}_0, \mathsf{k}_1) \leftarrow \mathcal{K}$
$\mathsf{ctxt}_0 \leftarrow \mathsf{Enc}(\mathsf{mpk}, 0, \mathsf{k}_0)$
$id_j := H_j(\mathsf{ctxt}_0)$, for all $j \in [k]$
$\mathsf{ctxt}_{id_j} \leftarrow \mathsf{Enc}(\mathsf{mpk}, id_j, \mathsf{k}_1)$
return $((\mathsf{ctxt}_0, (\mathsf{ctxt}_{id_j})_j), (\mathsf{k}_0, \mathsf{k}_1))$

$\underline{\mathsf{Decaps}(\mathsf{sk}_i, \mathsf{ctxt})}$:
$(T, (\mathsf{sk}'_{id})_{id\in[m]\cup\{0\}}) := \mathsf{sk}_i$
$(\mathsf{ctxt}_0, (\mathsf{ctxt}_{id_j})_{j\in[k]}) := \mathsf{ctxt}$
if $\mathsf{BF\text{-}Check}(H, T, \mathsf{ctxt}_0) = 1$ **return** $\bot$
find smallest $id \in [m]$ with $\mathsf{sk}'_{id} \neq \bot$
$\mathsf{k}_0 := \mathsf{Dec}(\mathsf{sk}'_0, \mathsf{ctxt}_0)$
$\mathsf{k}_1 := \mathsf{Dec}(\mathsf{sk}'_{id}, \mathsf{ctxt}_{id})$
if $\mathsf{k}_0 = \bot$ or $\mathsf{k}_1 = \bot$ **return** $\bot$
return $(\mathsf{k}_0, \mathsf{k}_1)$

Figure 2.14: BFKEM-IND-CPA-secure BFKEM scheme $\mathsf{BFKEM} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Punc}, \mathsf{Decaps})$ from IBE and BF.

the first part of a ciphertext (i.e., inserting the first part of the ciphertext as "tag" into the BF) which results in a secret key sk_n after n punctures. An (unpuctured) ciphertext ctxt_n, as freshly derived from $(\mathsf{ctxt}_n, \mathsf{k}_n) \leftarrow \mathsf{Encaps}(\mathsf{pk})$, yields $\mathsf{Decaps}(\mathsf{sk}_n, \mathsf{ctxt}_n) \neq \mathsf{k}_n$ if no IBE secret key is available anymore or an IBE decryption error occurs. Due to the bounded false-positive probability of BF, and the negligible key and correctness error $\epsilon(\lambda)$ and $\delta(\lambda)$ of IBE, this will happen with probability at most $(1 - e^{-\frac{(n+1/2)\cdot k}{m-1}})^k + \delta(\lambda) + \epsilon(\lambda)$ which yields Equation (2.1).

The following BFKEM-properties are mainly used in the security proof to achieve BFKEM-IND-CCA-secure BFKEMs from BFKEM-IND-CPA-secure BFKEMs via the FO transform [FO99] along the lines of the BFKEM-IND-CCA-proof given by Derler et al. [DJSS18, DGJ+21].

Extended correctness of BFKEM. According to Definition 2.5.4, we have to show (a) impossibility of false-negatives, (b) correctness of initial secret key, and (c) semi-correctness of punctured secret keys. For any number of secret-key punctures n, (a) holds due to the fact that sk_n (derived after puncturing on n ciphertexts) does not contain any IBE secret keys anymore which are capable of decrypting those ciphertexts due to the perfect completeness property of BF. (b) holds since sk_0 has all (initial) IBE secret keys to decrypt any honestly generated ciphertext correctly except with negligible probability due to IBE correctness with negligible decryption error $\delta(\lambda)$. Concerning (c), if decapsulation

does not fail with some (already punctured) secret key on some fixed ciphertext, i.e., there exists an IBE secret key to decrypt at least one ciphertext part, then Decaps outputs a key that is the same as the output of Decaps under sk_0 for that ciphertext except with negligible probability due to IBE correctness with negligible decryption error $\delta(\lambda)$.

Separable randomness of BFKEM. According to Definition 2.5.5, we show that $\mathsf{Encaps}(\mathsf{pk})$ can be written as $\mathsf{Encaps}(\mathsf{pk}; (r, (\mathsf{k}_0, \mathsf{k}_1)))$, for $(\mathsf{pk}, \cdot) \leftarrow \mathsf{KGen}(\lambda, m, k)$ and $(r, (\mathsf{k}_0, \mathsf{k}_1)) \leftarrow \mathcal{R} \times \mathcal{K}$ with randomness space $\mathcal{R} = \underbrace{\{0,1\}^\rho \times \cdots \times \{0,1\}^\rho}_{k \text{ times}}$, for large-enough integer ρ. We define $\mathsf{Encaps}(\mathsf{pk}; (r, (\mathsf{k}_0, \mathsf{k}_1)))$ as follows (see that the input $(\mathsf{pk}; (r, (\mathsf{k}_0, \mathsf{k}_1)))$ uniquely determines the output $((\mathsf{ctxt}_0, (\mathsf{ctxt}_{id_j})_j), (\mathsf{k}_0, \mathsf{k}_1))$:

$\underline{\mathsf{Encaps}(\mathsf{pk}; (r, (\mathsf{k}_0, \mathsf{k}_1)))}$:

$(\mathsf{mpk}, H) := \mathsf{pk}$ with $(H_j)_{j \in [k]} := H$
$(r_0, r_1, \ldots, r_k) := r$
$\mathsf{ctxt}_0 \leftarrow \mathsf{Enc}(\mathsf{mpk}, 0, \mathsf{k}_0; r_0)$
$id_j := H_j(\mathsf{ctxt}_0)$, for all $j \in [k]$
$\mathsf{ctxt}_{id_j} \leftarrow \mathsf{Enc}(\mathsf{mpk}, id_j, \mathsf{k}_1; r_j)$, for all $j \in [k]$
return $((\mathsf{ctxt}_0, (\mathsf{ctxt}_{id_j})_j), (\mathsf{k}_0, \mathsf{k}_1))$

Publicly-checkable puncturing of BFKEM. According to Definition 2.5.6, we have to show

$$\Pr\left[\mathsf{Decaps}(\mathsf{sk}_\ell, C) = \perp \not\Longleftrightarrow \mathsf{CheckPunct}(\mathsf{pk}, \mathcal{L}, C) = \perp\right] \leq \varepsilon(\lambda). \tag{2.2}$$

For $\ell \in \mathbb{N}$, we construct $\mathsf{CheckPunct}(\mathsf{pk}, \mathcal{L}, C)$, for $(\mathsf{pk}, \cdot) \leftarrow \mathsf{KGen}(\lambda, m, k)$ and any list of honestly generated ciphertexts $\mathcal{L} = (\mathsf{ctxt}_0, \ldots, \mathsf{ctxt}_{\ell-1})$ where sk_ℓ is punctured on, but *not* given as input to CheckPunct:

$\underline{\mathsf{CheckPunct}(\mathsf{pk}, \mathcal{L}, C)}$:

$(\mathsf{mpk}, H) := \mathsf{pk}$ with $(H_j)_{j \in [k]} := H$, $(C'_0, \ldots) := C$
$(\mathsf{ctxt}_0, \ldots, \mathsf{ctxt}_{\ell-1}) := \mathcal{L}$, for $\mathsf{ctxt}_i = (\mathsf{ctxt}_{i,0}, \ldots)$, for all $i \in [\ell]$
$T_i := \mathsf{BF\text{-}Update}(H, T_{i-1}, \mathsf{ctxt}_{i,0})$, for all $i \in [\ell]$
if $\mathsf{BF\text{-}Check}(H, T_\ell, C'_0) = 1$ **return** $\perp$
return $\not\perp$

See that CheckPunct runs in PPT since BF-Update and BF-Check are PPT algorithms. Furthermore, Decaps outputs $\perp$ if CheckPunct outputs $\perp$ while CheckPunct outputs $\perp$ if Decaps outputs $\perp$ except with negligible probability which is due to the negligible correctness error of IBE. Hence, Equation (2.2) follows.

γ-spreadness of BFKEM. See that the γ-spreadness property of the underlying IBE scheme directly carries over to the γ-spreadness property of BFKEM. Hence, if IBE is γ-spread, then BFKEM is γ-spread.

BFKEM-IND-CPA security of BFKEM. We start by showing the BFKEM-IND-CPA security of BFKEM.

Theorem 2.5.9. *If* IBE *is* IBE-sIND-CPA*-secure, then* BFKEM *is* BFKEM-IND-CPA*-secure. Concretely, for any PPT adversary $\mathcal{A}$ there is a PPT distinguisher D in the* IBE-sIND-CPA*-security experiment such that*

$$\mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cpa}}_{\mathsf{BFKEM},\mathcal{A}}(\lambda, m, k) \leq k \cdot m \cdot \mathsf{Adv}^{\mathsf{ibe\text{-}sind\text{-}cpa}}_{\mathsf{IBE},D}(\lambda). \tag{2.3}$$

Proof. We show the BFKEM-IND-CPA-security of BFKEM for any valid PPT adversary $\mathcal{A}$ in series of games where:

Hyb_0**:** This is the BFKEM-IND-CPA-security experiment.

Hyb_i **for** $i \in \{1, \ldots, k\}$**:** This is defined as Hyb_{i-1} except that the challenge-ciphertext element ctxt_{id_i} in ctxt^* associated to id_i is independent of the (challenge) bit b^*.

Hyb_{k+1} This is defined as Hyb_k except that the encapsulation key in the challenge ciphertext is independent of b^*.

We denote the event of the adversary winning hybrid Hyb_i as S_i. In hybdrid Hyb_{k+1}, $\mathcal{A}$ has no advantage (i.e., success probability of $\Pr[S_{k+1}] = 1/2$) in the sense of BFKEM-IND-CPA. We argue in hybrids that the Games $i \in [k+1]$ are computationally indistinguishable from Game 0.

Hybrids between Hyb_0 and Hyb_{k+1}. Each hybrid between Hyb_{i-1} and Hyb_i, $i \in [k]$, is constructed as follows:

- On input m and k, D samples $(H, T_0) \leftarrow \mathsf{BF\text{-}Gen}(m, k)$, for $H =: (H_j)_{j\in[k]}$ and sets $T_0 = 0^m$. Next, D samples (target identity) $id^* \leftarrow [m]$ and sends id^* to its IBE-sIND-CPA-challenger. D retrieves mpk in return and sets $\mathsf{pk} := (\mathsf{mpk}, H)$.
- For all $id \in ([m] \cup \{0\}) \setminus \{id^*\}$, D retrieves $\mathsf{sk}_0 := (\mathsf{sk}_{id})_{id}$ from its Ext-oracle. (Note that D does not have a secret key for id^*.) Looking ahead, with significant probability, D will prepare a challenge ciphertext for $\mathcal{A}$ that will include the IBE challenge ciphertext retrieved from the IBE-sIND-CPA-challenger for id^*. In that sense, $\mathcal{A}$ has to query the overall challenge ciphertext to the Punc'-oracle if $\mathcal{A}$ wants to receive a secret key via the Cor-oracle, which results in "deleting" the secret key for id^* and not providing it to $\mathcal{A}$. Since D does not possess the secret key for id^*, it does not need to prepare a query answer for $\mathcal{A}$ that includes a secret key for id^*. Given that, all Cor-queries can be answered correctly.

- D sends $\mathsf{k}_1^{(0)}, \mathsf{k}_1^{(1)} \leftarrow \mathcal{M}$ to its IBE-sIND-CPA-challenger and retrieves $\mathsf{ctxt}^*_{id^*} \leftarrow \mathsf{Enc}(\mathsf{mpk}, id^*, \mathsf{k}_1^{(b)})$, for some (unknown) $b \leftarrow \{0,1\}$.
- D samples $b^* \leftarrow \{0,1\}$, computes $\mathsf{ctxt}_0 \leftarrow \mathsf{Enc}(\mathsf{mpk}, 0, \mathsf{k}_0)$, for $\mathsf{k}_0 \leftarrow \mathcal{M}$, and sets $(id_j)_j := (H_j(\mathsf{ctxt}_0))_{j\in[k]}$. If $id_i \neq id^*$, D "aborts" and sends b^* to its IBE-sIND-CPA-challenger. (See that D aborts with probability $(m-1)/m$.) Otherwise, D prepares:

  Part ciphertexts $1, \ldots, i-1$: $\mathsf{ctxt}_{id_j} \leftarrow \mathsf{Enc}(\mathsf{mpk}, id_j, \mathsf{k}_1^{(1)})$, for all $(id_j)_{j\in[i-1]}$.

  Part ciphertext i: $\mathsf{ctxt}_{id_i} := \mathsf{ctxt}^*_{id^*}$.

  Part ciphertexts $i+1, \ldots, k$: $\mathsf{ctxt}_{id_j} \leftarrow \mathsf{Enc}(\mathsf{mpk}, id_j, \mathsf{k}_1^{(0)})$, for all $(id_j)_{j\in[k]\setminus[i]}$.

- D sends $(\mathsf{pk}, \mathsf{ctxt}^* := (\mathsf{ctxt}_0, (\mathsf{ctxt}_{id_j})_j), \mathsf{k})$ to A, for $\mathsf{k} := (\mathsf{k}_0, \mathsf{k}_1^{(0)})$ if $b^* = 0$ and $\mathsf{k} := (\mathsf{k}_0, \mathsf{k}_1^{(1)})$ if $b^* = 1$.
- $\mathcal{A}$ has access to a $\mathsf{Punc}'(\mathsf{ctxt})$-oracle which runs $\mathsf{sk}_{i+1} \leftarrow \mathsf{Punc}(\mathsf{sk}_i, \mathsf{ctxt})$ for each invocation $i = 0, 1, \ldots, q$ and sets $\mathcal{L} := \mathcal{L} \cup \{\mathsf{ctxt}\}$ for initially empty set $\mathcal{L}$ and number of queries q to Punc. The Cor-oracle returns sk_i iff $\mathsf{ctxt}^* \in \mathcal{L}$, for some query $i \in [q]$.
- Eventually, $\mathcal{A}$ outputs a guess b' which D forwards as $b' \oplus b^*$ to its IBE-sIND-CPA-challenger.

In the hybrid between Hyb_k and Hyb_{k+1}: proceed as in Hyb_k, but send $(\mathsf{pk}, \mathsf{ctxt}^* := (\mathsf{ctxt}_0, (\mathsf{ctxt}_{id_j})_j), (\mathsf{k}_0, \mathsf{k}'_1))$, for uniform $\mathsf{k}'_1 \leftarrow \mathcal{M}$ to $\mathcal{A}$.

Analysis. In the hybrids between Hyb_{i-1} and Hyb_i, for each $i \in [k]$, we have that if the IBE challenge ciphertext is associated to $b = 0$, then we are in Hyb_{i-1}; otherwise, if $b = 1$, then we are in Hyb_i.

In the hybrid between Hyb_k and Hyb_{k+1}, the change is information-theoretic, i.e., the challenge ciphertext encapsulates a uniformly random key-element $\mathsf{k}_1^{(1)}$ and the second part of the encapsulation key k'_1 is sampled uniformly at random which yields $\Pr[S_{k+1}] = 1/2$. (See that any adversary can always retrieve k_0 as it can always decrypt ctxt_0 if it queries the Cor-oracle to receive any secret key after querying ctxt^* to Punc'.)

Moreover, in each hybrid between Hyb_{i-1} and Hyb_i, for each $i \in [k]$, we have that $\Pr[id_i = id^*] = 1/m$ and D is a PPT algorithm. Putting things together, for k game hops, we conclude that Equation (2.3) holds. □

BFKEM-IND-CCA security of BFKEM′. We construct a slight variant of our BFKEM scheme, dubbed BFKEM′, via the FO transform [FO99] along the lines of Derler et al. [DJSS18, DGJ+21]. We want to mention that the FO transform does not work generically for any BFKEM-IND-CPA-secure BFKEM and no generic framework as in the case of KEMs exists. Hence, we consider the direct product compiler in Section 2.5.1 and

$\mathsf{KGen}'(\lambda, m, k)$:	return $(\mathsf{pk}, \mathsf{sk}_0) \leftarrow \mathsf{KGen}(\lambda, m, k)$.
$\mathsf{Encaps}'(\mathsf{pk})$:	on input pk, sample $\mathsf{k} \leftarrow \mathcal{K}$, compute $(r, \mathsf{k}') := \mathsf{G}(\mathsf{k}) \in \{0,1\}^{k\cdot\rho+\lambda}$ and $(\mathsf{ctxt}, \mathsf{k}) \leftarrow \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k}))$, and return $(\mathsf{ctxt}, \mathsf{k}')$.
$\mathsf{Punc}'(\mathsf{sk}_{i-1}, \mathsf{ctxt})$:	return $\mathsf{sk}_i \leftarrow \mathsf{Punc}(\mathsf{sk}_{i-1}, \mathsf{ctxt})$.
$\mathsf{Decaps}'(\mathsf{sk}_i, \mathsf{ctxt})$:	on input secret key sk_i and ciphertext ctxt, compute $\mathsf{k} \leftarrow \mathsf{Decaps}(\mathsf{sk}_i, \mathsf{ctxt})$ and return $\perp$ if $\mathsf{k} = \perp$. Otherwise, compute $(r, \mathsf{k}') := \mathsf{G}(\mathsf{k})$ and return k' if $(\mathsf{ctxt}, \mathsf{k}) = \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k}))$, else output $\perp$.

Figure 2.15: BFKEM-IND-CCA-secure BFKEM′ from BFKEM-IND-CPA-secure BFKEM and hash function G (modeled as random oracle (RO) in the security proof).

the general proof methodology as given in [DJSS18, DGJ+21] to achieve BFKEM-IND-CCA security for BFKEM′. Furthermore, [DJSS18, DGJ+21] requires perfect correctness for unpunctured keys which our BFKEM definition cannot guarantee. Hence, we have to reprove the BFKEM-IND-CCA security for BFKEM′, although the proof techniques are almost the same as presented in [DJSS18, DGJ+21].

We construct a BFKEM-IND-CCA-secure BFKEM as follows. Let $\mathsf{BFKEM} = (\mathsf{KGen}, \mathsf{Encaps}, \mathsf{Punc}, \mathsf{Decaps})$ be a BFKEM-IND-CPA-secure BFKEM scheme with key space $\mathcal{K}$ and non-negligible correctness error $\delta = \delta(\lambda, m, k, n)$. Furthermore, let BFKEM have the extended correctness, separable randomness, publicly-checkable puncturing, and γ-spreadness properties. We construct a BFKEM-IND-CCA-secure BFKEM scheme $\mathsf{BFKEM}' = (\mathsf{KGen}', \mathsf{Encaps}', \mathsf{Punc}', \mathsf{Decaps}')$ with key space $\mathcal{K}' = \{0,1\}^\lambda$ using a variant of the FO transform in Figure 2.15 (let $\mathsf{G}\colon \mathcal{K} \to \{0,1\}^{k\cdot\rho+\lambda}$, for BFKEM parameter k and large-enough integer ρ, be a hash function modeled as random oracle (RO) in the security proof).

See that correctness (Definition 2.5.3) directly carries over from BFKEM to BFKEM′, i.e., it is straightforward to verify that if BFKEM is correct then BFKEM′ is correct. (We only argue to achieve the correctness property together with BFKEM-IND-CCA-security for BFKEM′ here since the other BFKEM properties are essentially only needed for the FO transform starting with a BFKEM-IND-CPA-secure BFKEM having those other properties as well.)

Theorem 2.5.10. *If a BFKEM* BFKEM *is* BFKEM-IND-CPA*-secure with the (extended) correctness, separable randomness, publicly-checkable puncturing, and γ-spreadness properties, then* BFKEM′ *is* BFKEM-IND-CCA*-secure. Concretely, for any PPT adversary $\mathcal{A}$ making at most $q_\mathsf{G} = q_\mathsf{G}(\lambda)$ queries to the random oracle* G *and negligible $\delta = \delta(\lambda)$, there is a distinguisher $\mathcal{D}$ in the* BFKEM-IND-CPA*-security experiment such that*

$$\mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cca}}_{\mathsf{BFKEM}',\mathcal{A}}(\lambda, m, k) \leq \mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cpa}}_{\mathsf{BFKEM},\mathcal{D}}(\lambda, m, k) + 3 \cdot \delta + \frac{q_\mathsf{G}}{2^\gamma}.$$

Since the proof methodology is almost the same as presented in [DJSS18, DGJ+21], we refer the reader to Appendix A.2.4 for the proof. Essentially, we deviate from [DJSS18,

DGJ+21] such that the adapted BFKEM-properties extended correctness, separable randomness, and publicly-checkable puncturing have to be carefully integrated into the game hops which — instead of a perfectly indistinguishable game hops in [DJSS18, DGJ+21] — we rely on negligibly indistinguishable game hops by using slightly adapted properties for BFKEM.

On the instantiation of BFKEM′ from lattice- and code-based IBE schemes. For a BFKEM-IND-CCA-secure BFKEM′, we require the underlying BFKEM to be BFKEM-IND-CPA-secure *and* have the properties extended correctness, separable-randomness, publicly-checkable puncturing, and γ-spreadness. Since we build CPA-secure BFKEMs from selectively CPA-secure IBEs, we require by any potential lattice- or code-based IBE to have a (non-)negligible correctness error (in the sense of HHK [HHK17a]) and the property of γ-spreadness. (See that the properties separable-randomness and publicly-checkable puncturing for a BFKEM-IND-CPA-secure BFKEM can be shown without any requirements on the underlying IBE. Furthermore, extended correctness holds with respect to a negligible correctness error of the underlying IBE.) Natural candidates for lattice- and code-based selectively CPA-secure IBEs are the schemes of Agrawal, Boneh, and Boyen (ABB) [ABB10] or Ducas, Lyubashevsky, and Prest [DLP14] (i.e., lattice-based IBEs) and the approach due to Gaborit et al. (GHPT) (i.e, a code-based IBE) [GHPT17] (considering the changes from [DT18]). We note though that correctness in the sense of HHK [HHK17a] has not been studied for those IBE schemes and a rigorous study of correctness for code- and lattice-based IBEs is an interesting direction for future research. For GHPT in particular, we expect that correctness and γ-spreadness can be lifted from the underlying PKE, RankPKE, as ciphertexts of the IBE are RankPKE-ciphertexts whereas a part of the public key is identity-dependent.

Table 2.4: Sizes of BFKEM when instantiated with GVP or GHPT.

IBE	assumption	sk	pk	ctxt
GVP-80	lattice-based	19.21 GB	1.62 KB	17.46 KB
GVP-192	lattice-based	47.15 GB	3.78 KB	40.28 KB
GHPT-128	code-based	643.73 GB	252 KB	215.79 MB
Boneh-Franklin [DJSS18]	pairing-based	717.18 MB	95.5 B	255.5 B

2.5.3 Comparison of BFKEM Instantiations

To instantiate a BFKEM from post-quantum IBE schemes, we investigate instantiations based on a selectively IND-CPA-secure lattice-based or code-based IBEs. As far as lattices are concerned, the first such construction was [GPV08] after which numerous others followed [ABB10, CHKP10, DLP14, ZCZ16]. To compute the dimension of a lattice-based BFKEM, we start from the GVP-IBE instantiation of [DLP14], for which an implementation and concrete dimensions were given for 80 and 192-bit quantum security.

We set the parameter of the BFKEM as in [DJSS18], i.e., targeting the maximum number of allowed punctures to $n = 2^{20}$, which amounts to adding 2^{12} elements per day to the BF for a year, and allowing for a false-positive probability of 10^{-3}, we obtain $m = 1.5 \cdot 10^7$ and $k = 10$. A similar procedure can be applied to the code-based IBE of Gaborit et al. (GHPT) [GHPT17] achieving 128-bit quantum security. We note though that with recent advances in the cryptanalysis, these instances may provide less security.[6] Also, we note that for obtaining a BFKEM-IND-CCA-secure BFKEM, the respective IBE needs to satisfy correctness in the sense of HKK (which, as mentioned before, one would have to assume as it has not been studied before). Table 2.4 provides an overview including the pairing-based BFKEM from [DJSS18]. For the latter, we assume the use of the pairing-friendly BLS12-381 curve with 120-bit classical security.

[6] In particular, due to an attack by Debris-Alazard and Tillich in [DT18] on GHPT a concrete choice of secure parameters is unclear.

CHAPTER 3

Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable

Abstract

A succinct non-interactive argument of knowledge (SNARK) allows a prover to produce a short proof that certifies the veracity of a certain NP-statement. In the last decade, a large body of work has studied candidate constructions that are secure against quantum attackers. Unfortunately, no known candidate matches the efficiency and desirable features of (pre-quantum) constructions based on bilinear pairings.

In this work, we make progress on this question. We propose the first lattice-based SNARK that simultaneously satisfies many desirable properties: It (i) is tentatively post-quantum secure, (ii) is publicly-verifiable, (iii) has a logarithmic-time verifier and (iv) has a purely algebraic structure making it amenable to efficient recursive composition. Our construction stems from a general technical toolkit that we develop to translate pairing-based schemes to lattice-based ones. At the heart of our SNARK is a new lattice-based vector commitment (VC) scheme supporting openings to constant-degree multivariate polynomial maps, which is a candidate solution for the open problem of constructing VC schemes with openings to beyond linear functions. However, the security of our constructions is based on a new family of lattice-based computational assumptions which naturally generalises the standard Short Integer Solution (SIS) assumption.

This chapter presents the first result of the collaboration with Martin R. Albrecht, Russell W. F. Lai, Giulio Malavolta and Sri AravindaKrishnan Thyagarajan and was published at the 42nd Annual International Cryptology Conference (CRYPTO'22) under the title "Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable

- (Extended Abstract)" [ACL+22]. I am manly responsible for the security proof and efficiency analysis of the functional commitment scheme. Further, I contributed to the design and security analysis of the adaptor signature construction. I am also responsible for writing the corresponding sections of the chapter. The accompanying appendix contains omitted constructions and proofs.

3.1 Introduction

A succinct non-interactive argument of knowledge (SNARK) [Kil92, Mic94] allows a prover to convince a verifier that they know a witness to an NP statement. The succinctness property demands that the size of the proof and, after preprocessing, the work of the verifier are sublinear in (ideally independent of) the time needed to check the validity of the witness. Over the last decade, SNARKs have witnessed a meteoric rise in their efficiency and applicability [BCG+13, BCTV14b, PHGR13, BCC+09, CG08, GGM14]. More recently, SNARKs have found their way into real-world systems in the context of blockchain-based cryptocurrencies [BCG+14, KMS+16, BGH19, BDFG21a, BMRS20].

The looming threat of quantum computers has given rise to a movement in the cryptographic community to investigate cryptographic constructions from assumptions that would plausibly withstand the presence of a quantum attacker. Unfortunately, present SNARKs based on post-quantum assumptions are in many ways inferior to pre-quantum constructions based on bilinear pairings. The goal of this work is to make progress in this area.

3.1.1 The Seascape of SNARKs

To put our work into context, we give a brief outline of the current seascape of SNARK constructions. We split the schemes depending on the underlying cryptographic assumptions used as the source of hardness.

Bilinear Pairings. To date, the most efficient and feature-rich SNARKs are constructed over bilinear pairing groups (e.g. [Gro16]) with a trusted setup. Typically, a pairing-based SNARK proof consists of only a small constant number of base group elements and is also publicly verifiable. Furthermore, offline preprocessing can often be performed, such that the online verification time is sublinear in the size of the statement being proved and the corresponding witness. Moreover, pairing-based SNARKs are favourable because of their algebraic structures that is known to enable proof batching [LMR19, BMM+21] and efficient recursive composition [BCTV14a]. However, due to their reliance on the hardness of problems related to discrete logarithms, pairing-based SNARKs are not sound against a cheating quantum prover.

Random Oracles. Promising post-quantum candidate for SNARKs are constructions based on Micali's CS proofs paradigm: They are obtained by first building an interactive argument using (generalisations of) probabilistically checkable proofs (PCP) [Kil92], then

compiling it into a non-interactive one using the Fiat-Shamir transformation [FS87] in the random oracle (RO) model.

A major difference between pairing-based and RO-based SNARKs, from both theoretical and practical perspectives, is the algebraic structure of the verification algorithm. In RO-based SNARKs, the verification algorithms query the RO, which is a combinatorial object. This is especially important when recursively composing the SNARK: On the theoretical side, proving the knowledge of a valid RO-based SNARK proof requires specifying the circuit computing the RO. This makes it challenging to formally argue about soundness, even in the RO model. From a practical perspective, the RO is instantiated with cryptographic hash functions, which typically have high multiplicative degree.[1] Since the multiplicative degree of the relation being proven often dominates the prover computation complexity in SNARKs, proving the satisfiability of a cryptographic hash function becomes computationally expensive.

Lattices. A prominent source of hardness for post-quantum security are computational problems over lattices. Not only do lattice-based assumptions allow us to build most standard cryptographic primitives, e.g. [Reg05, GPV08], but also enable new powerful primitives [Gen09, GVW15a, WZ17, GKW17], which are currently out of the reach of group-based assumptions. Unfortunately, in the context of SNARKs, lattices have yet to be established as competitive alternatives to group-based constructions. So far, lattice-based SNARKs either require designated verifiers [GMNO18, ISW21] or linear-time verification [ACK21, BCS21].

Beyond their theoretical appeal, one additional motivation for constructing lattice-based SNARKs is that they are potentially more compatible with other basic lattice-based primitives when composing them to construct more advanced systems. More concretely, consider the task of proving the satisfiability of certain algebraic relations over a ring $\mathcal{R}$ by a solution vector of norm bounded by some δ, a language which arises naturally when composing lattice-based building blocks. Using an argument system for proving algebraic relations over a finite field without norm constraints, arithmetisation would be needed to express certain witness component in, say, binary representation and translate the bounded-norm condition to the satisfiability of a potentially-high-degree polynomial, depending on the choice of the norm and the norm bound δ. In contrast, the bounded-norm constraint could be proven natively if we have an argument system which supports proving the satisfiability of algebraic relations over $\mathcal{R}$ by solutions of norm bounded by some $\alpha \leq \delta$. This is done by expressing the solution vector in a likely more compact $O(\alpha)$-ary representation such that, if the representation has norm bounded by α, then the original solution has norm bounded by δ.

[1] Though we mention that there is recent progress [ARS+15, GKK+19] in crafting hash functions that are friendlier to multiparty computation and argument systems.

3.1.2 Our Contributions

In this work, we construct the first lattice-based SNARK for an NP-complete language defined over a ring $\mathcal{R}$. Specifically, the language being supported is the satisfiability of polynomial maps over $\mathcal{R}$ by bounded-norm solutions. Our construction qualitatively matches pairing-based SNARKs, i.e. it is publicly verifiable and can achieve sublinear verification time given preprocessing, while requiring a trusted setup. In addition, it is tentatively post-quantum secure. Furthermore, our construction uses only algebraic operations over a ring $\mathcal{R}$, and is therefore friendly to recursive composition. The soundness of our scheme is based on new lattice-based (knowledge) assumptions. The introduction of new knowledge assumptions is, to some extent, necessary: The work of Gentry and Wichs [GW11] shows that the soundness of any SNARK cannot be based on falsifiable assumptions in a black-box manner. We summarise the main steps of our work in the following.

(1) Translation Technique. We put forward a new paradigm for translating pairing-based constructions to the lattice world. Our constructions stem from techniques from the literature on pairing-based cryptography [LY10], while simultaneously exploiting the ring structure offered by the lattice setting. We develop the necessary technical toolkit that helps us mimic operations of pairing-based VC constructions in the lattice setting. We view this translation strategy as a major conceptual contribution of our work and we expect it to be instrumental in enabling new applications of lattice-based cryptography.

(2) Vector Commitments for Constant-Degree Polynomials. A vector commitment (VC) allows a committer to commit to a vector of w values $\mathbf{x} \coloneqq (x_0, \ldots, x_{w-1}) \in \mathcal{R}^w$ and then reveal selected portions of the input vector, or more generically a function $f : \mathcal{R}^w \to \mathcal{R}^t$ over the input vector, along with a proof π that can be publicly verified. We require both the commitment and the opening proof to be *compact*. In terms of security, we want to ensure an adversary cannot output a valid opening proof for an incorrect function evaluation of the input vector. VCs have been established as a central primitive in cryptography [CF13, LRY16, Fis19, LM19, GRWZ20, CFG+20]. As a central technical contribution, we present the first (lattice-based) VC that supports openings beyond linear functions. Specifically, our VC commits to short vectors of ring elements $\mathbf{x} \in \mathcal{R}^w$ and supports openings to constant-degree d multivariate polynomial maps. We then show how this VC is sufficient to construct SNARKs for the satisfiability of degree-d polynomial maps (which is NP-complete for $d \geq 2$) by bounded-norm solutions.

(3) New Assumptions and Analysis. Our translation techniques (and consequently the resulting cryptographic schemes) rely on a new family of assumptions that we refer to as the *k-Ring-Inhomogenous Short Integer Solution* (or k-R-ISIS for short) assumptions. Roughly, a k-R-ISIS assumption says that it is hard to find a short preimage $\mathbf{u}_{g^*}$ satisfying $\langle \mathbf{a}, \mathbf{u}_{g^*} \rangle = g^*(\mathbf{v}) \bmod q$, where g^* is a Laurent monomial[2] and $\mathbf{v}$ is a random point,

[2] A Laurent monomial is a monomial where negative powers are allowed. Generally, one could consider k-R-ISIS problems for rational functions.

given short preimages of other Laurent monomials $\mathcal{G}$ evaluated on the same random point. Our new assumptions can be viewed as inhomogenous ring variants of the k-SIS assumption [BF11, LPSS14] (where the rational functions are zeros). The key difference to k-SIS is that we allow to hand out more preimages than the dimension of $\mathbf{a}$ but these preimages are all of different images.

In fact, the assumptions we introduce, k-M-ISIS, are slightly more general in being defined over modules rather than rings. Our generalisation to modules is motivated by the knowledge assumptions that we also introduce. In the knowledge assumptions images live in a moderately sized submodule.

We consider the introduction and study of the k-R-ISIS assumptions as a contribution to the programme of charting the territory between LWE and multilinear maps assumptions called for in [Agr20].

To gain confidence in our newly introduced assumptions, we initiate their study. We show that certain subclasses of the k-R-ISIS problems (parameterised by the algebraic structure on the k-R-ISIS images) are as hard as the R-SIS problem. We show that, as expected, the k-M-ISIS problems are as hard as their k-R-ISIS counterparts, although the former have slightly skewed parameters. We also show that certain k-M-ISIS problems are as hard as the k-M-SIS problem, the natural module variant of the k-SIS problem, where the former have higher module ranks. Furthermore, we show that the k-M-ISIS problems for $(\mathcal{G}, g^*)$ is as hard as those for $(\mathcal{G}, 0)$, and that the hardness is preserved when scaling both $\mathcal{G}$ and g^* multiplicatively by any non-zero Laurent monomial.

However, since none of the reductions from well-established problems cover the case we rely upon in our constructions, we perform cryptanalysis to assess the hardness of general k-M-ISIS problems. While we did not identify any structural weaknesses, we encourage independent analysis to gain confidence in or invalidate our assumptions.

(4) Post-Quantum Security. As a contribution of independent interest, we show that our VC satisfies a strong notion of binding known as *collapsing* (as an ordinary commitment, not with respect to functional openings), a recently introduced security notion in the quantum setting [Unr16]. For this, we introduce a new technique of embedding NTRU ciphertexts into the public parameters of our VC. To the best of our knowledge, this is the first VC not based on Merkle trees that is shown to satisfy such a notion.

(5) New Applications. Our SNARK supports proving the satisfiability of polynomial maps over $\mathcal{R}$ by bounded-norm solutions, a language which directly captures those statements which naturally arise in lattice-based cryptographic constructions. We highlight two native applications of our SNARK which do not rely on expensive conversions between different NP-complete languages.

The first application is the recursive composition of our SNARK, which refers to the process of using the SNARK to prove knowledge of another SNARK proof and the satisfiability of a polynomial map; for details see 3.6.2. This is natively supported because

the verification algorithm of our SNARK construction is itself checking the satisfiability of certain algebraic relations over $\mathcal{R}$ by a bounded-norm solution. Recursive composition of SNARKs is a general purpose technique for aggregating proofs or proving complex statements in a piece-by-piece fashion. The technique is also useful for constructing incremental verifiable computation [Val08] and verifiable delay functions [BBBF18, Gro21].

The second application is the aggregation of GPV signatures [GPV08]. While it is folklore that any signatures can be aggregated by a SNARK for an NP-complete language, we stress that the GPV verification algorithm, again, checks the satisfiability of certain algebraic relations over $\mathcal{R}$ by a bounded-norm solution which our SNARK natively supports. We discuss how to handle relations in $\mathcal{R}_q$ in Section 3.6.1. Apart from obtaining short aggregated GPV signatures, in the setting where a set of n signers are signing a common message at a time, the verification of the aggregated signatures could be preprocessed, resulting in an online verification time *sublinear* in n. As a bonus result on GPV signatures, we further show how to construct lattice-based adaptor signatures [AEE+21] based on the GPV paradigm. Combining the two results, we obtain the first aggregatable adaptor signatures.

Open Problems. Our work paves the way for what we believe to be an exciting line of research. As we initiate the study of inhomogenous variants of the k-SIS assumptions, we ask whether better (possibly quantum) algorithms can be found for solving this problem that exploit the additional algebraic structure. We also presume that for further families of rational functions the k-R-ISIS assumption can be shown to be as hard as standard hard lattice problems. Another compelling question is to study new cryptographic applications of the k-R-ISIS family. We expect that such an abstraction will be useful in transferring techniques from pairing-based cryptography into the lattice world.

3.1.3 Technical Overview

We give a concise overview of the process of obtaining our lattice-based SNARK.

From Vector Commitments to SNARKs. In this work, we are interested in VCs supporting openings to constant-degree-d w-variate t-output polynomial maps with bounded coefficients. The standard properties of interest for VCs are:

Compactness. Commitments and opening proofs are of size $\mathsf{poly}(\lambda, \log w, \log t)$.

Binding. It is infeasible to produce a commitment c and proofs for polynomials maps, such that the system of equations induced by them is not satisfiable.[3]

In addition, we require the following stronger notion of binding.

[3] This generalises position binding.

Extractability. To produce a commitment c and a proof that the image of a polynomial map f at the committed vector is $\mathbf{y}$, one must know a preimage $\mathbf{x}$ such that c is a commitment of $\mathbf{x}$ and $f(\mathbf{x}) = \mathbf{y}$.

It is well known that one can construct SNARKs from VCs supporting linear openings in the RO model [LM19]. However, in this work we take a different route and adopt a more structured approach to construct SNARKs. Specifically, recall that the satisfiability of systems of degree-d polynomials is NP-complete for any constant $d \geq 2$. As such, a SNARK can be trivially constructed from a compact and extractable VC for degree-d polynomials: The prover simply commits to the root of the system $(f, \mathbf{y})$ and immediately produces an opening proof for $(f, \mathbf{y})$. As a concrete example, a popular NP-complete language supported by existing SNARKs is rank-1 constraint satisfiability (R1CS). An R1CS instance consists of three matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ over a field or in general a ring. The instance is satisfied by a vector $\mathbf{x}$ if $(\mathbf{A} \cdot (1, \mathbf{x})) \circ (\mathbf{B} \cdot (1, \mathbf{x})) = (\mathbf{C} \cdot (1, \mathbf{x}))$, where $\circ$ denotes the Hardamard product. It is easy to see that an R1CS instance is a special case of an instance $(f, \mathbf{y})$ of degree-2 polynomial satisfiability where $f(\mathbf{X}) := (\mathbf{A} \cdot (1, \mathbf{X})) \circ (\mathbf{B} \cdot (1, \mathbf{X})) - (\mathbf{C} \cdot (1, \mathbf{X}))$ and $\mathbf{y} = \mathbf{0}$. For a full description of our SNARK we refer the reader to Section 3.6.

Throughout the rest of this overview, we therefore focus on constructing lattice-based VCs supporting degree-d openings. Since known constructions are restricted to positional openings, we turn our attention to pairing-based schemes (which support linear openings) and develop a new strategy to translate them into lattice-based VCs and simultaneously to extend the degree to $d > 1$.

General Translation Strategy. Our strategy for constructing a lattice-based VC is a novel translation technique that lets us port techniques from the pairing-land to the lattice-land. We describe a general translation strategy for translating not only VC but also potentially other pairing-based constructions to the lattice setting. For the group setting, we adopt the implicit notation for bilinear groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_t$ of prime order q, i.e. the vector of elements in $\mathbb{G}_i$ with (entry-wise) discrete logarithm $\mathbf{x} \in \mathbb{Z}_q$ base an arbitrary fixed generator of $\mathbb{G}_i$ is denoted by $[\mathbf{x}]_i$, with group operations written additively, and the pairing product between $[\mathbf{x}]_1$ and $[\mathbf{y}]_2$ is written as $\langle [\mathbf{x}]_1, [\mathbf{y}]_2 \rangle$. For the lattice setting, we let $\mathcal{R}$ be a cyclotomic ring, $q \in \mathbb{N}$ be a large enough rational prime such that random elements in $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$ are invertible with non-negligible probability.

Consider a pairing-based construction where the elements $\{ [1]_1, [g(\mathbf{v})]_t \}_{g \in \mathcal{G}}$ are publicly available to all parties, where $\mathcal{G}$ is a set of linearly-independent rational functions and $\mathbf{v}$ is a vector of secret exponents. An authority, knowing the secret exponents $\mathbf{v}$, is responsible for giving out secret elements $\{ [g(\mathbf{v})]_2 \}_{g \in \mathcal{G}}$ to user A. In turn, user A can compute $[u]_2 := \sum_{g \in \mathcal{G}} c_g \cdot [g(\mathbf{v})]_2$ and present it to user B, who can then check the correctness of $[u]_2$ by checking

$$\langle [1]_1, [u]_2 \rangle \stackrel{?}{=} \sum_{g \in \mathcal{G}} c_g \cdot [g(\mathbf{v})]_t.$$

Note that in this check one side of the pairing (i.e. $[1]_1$) is public, while the other side (i.e. $[u]_2$) is computed from secrets delegated by the authority to user A. This property will be crucial for our translation technique to apply.

The above structure can be seen in many pairing-based constructions. For example, the secret vector $\mathbf{v}$ could be a trapdoor, a master secret key of an identity-based encryption scheme, or a signing key; the delegated secrets $\{ [g(\mathbf{v})]_2 \}_{g\in\mathcal{G}}$ could be hints given alongside the public parameters of a VC, an identity-based secret key, or a signature; and the pairing-product check could be for opening proof verification, decryption, or signature verification.

Our strategy of translating the above to a lattice-based construction is as follows. First, the public elements $\{ [1]_1, [g(\mathbf{v})]_t \}_{g\in\mathcal{G}}$ over $\mathbb{G}_1$ and $\mathbb{G}_t$ are translated to the public vector and elements $\{ \mathbf{a},\ g(\mathbf{v}) \}_{g\in\mathcal{G}}$, where $\mathbf{a}$ and $\mathbf{v}$ are random vectors over $\mathcal{R}_q$ and $\mathcal{R}_q^\times$ respectively. Since $\{ g(\mathbf{v}) \}_{g\in\mathcal{G}}$ does not necessarily hide $\mathbf{v}$ in the lattice setting (e.g. when $\mathcal{G}$ consists of many linear functions), the authority might as well publicly hand out the vectors $\{ \mathbf{a}, \mathbf{v} \}$ directly. Next, the secret elements $\{ [g(\mathbf{v})]_2 \}_{g\in\mathcal{G}}$ are translated to the *short* secret vectors $\{ \mathbf{u}_g \}_{g\in\mathcal{G}}$ satisfying $\langle \mathbf{a}, \mathbf{u}_g \rangle = g(\mathbf{v}) \bmod q$. These short preimages can be sampled given a trapdoor of $\mathbf{a}$, which the authority should have generated alongside $\mathbf{a}$. Given $\{ \mathbf{u}_g \}_{g\in\mathcal{G}}$, user A can similarly compute $\mathbf{u} \coloneqq \sum_{g\in\mathcal{G}} c_g \cdot \mathbf{u}_g$, although the coefficients c_g are now required to be short. The pairing-product check is then translated to checking

$$\langle \mathbf{a}, \mathbf{u} \rangle \stackrel{?}{\equiv} \sum_{g\in\mathcal{G}} c_g \cdot g(\mathbf{v}) \bmod q \qquad \text{and} \qquad \mathbf{u} \text{ is short.}$$

The same strategy can also be used to translate (conjectured-)hard computational problems over bilinear groups to the lattice setting to obtain also seemingly-hard problems. For example, consider a variant of the ℓ-Diffie-Hellman Exponent problem, which asks to find $[v^\ell]_2$ given $([1]_1, [1]_2, [v]_2, \ldots, [v^{\ell-1}]_2)$. A natural lattice-counterpart of the problem is to find a short preimage $\mathbf{u}_\ell$ satisfying $\langle \mathbf{a}, \mathbf{u}_\ell \rangle \equiv v^\ell \bmod q$ given short preimages $(\mathbf{u}_i)_{i\in\mathbb{Z}_\ell}$ each satisfying $\langle \mathbf{a}, \mathbf{u}_i \rangle = v^i \bmod q$.

We remark that a direct translation of pairing-based constructions does not necessarily yield the most efficient lattice-based scheme. For this reason, it will be useful to generalise pairing-based constructions into a family parameterised by the function class $\mathcal{G}$. We will then have the freedom to pick $\mathcal{G}$ to optimise the efficiency of translated lattice-based scheme.

Translating Vector Commitments. We next demonstrate how the above translation strategy can be applied to translate pairing-based VCs, using the following pairing-based VC with openings to linear forms $f : \mathbb{Z}_q^w \to \mathbb{Z}_q$ adapted from [CF13, LRY16, LM19] as an example.

- Public parameters: $\left([1]_1, [1]_2, ([v_i]_1)_{i\in\mathbb{Z}_w}, ([\bar{v}_j]_2)_{j\in\mathbb{Z}_w}, ([v_i \cdot \bar{v}_j]_2)_{i,j\in\mathbb{Z}_w : i\neq j}, [\bar{v}]_t\right)$ where $\bar{v} = \prod_{k\in\mathbb{Z}_w} v_k$ and $\bar{v}_j = \bar{v}/v_j$.

- Committing $\mathbf{x} \in \mathbb{Z}_q$: $[c]_1 \coloneqq \sum_{i \in \mathbb{Z}_w} x_i \cdot [v_i]_1 = \langle [\mathbf{v}]_1, \mathbf{x} \rangle$
- Opening f : $[u]_2 \coloneqq \sum_{i,j \in \mathbb{Z}_w : i \neq j} f_j \cdot x_i \cdot [v_i \cdot \bar{v}_j]_2$
- Verifying (f, y): $\langle [1]_1, [u]_2 \rangle \stackrel{?}{=} \left\langle [c]_1, \sum_{j \in \mathbb{Z}_w} f_j \cdot [\bar{v}_j]_2 \right\rangle - y \cdot [\bar{v}]_t$

The weak binding property of the scheme, i.e. the infeasibility of opening a commitment c to both (f, y) and (f, y') with $y \neq y'$, relies on the hardness of computing $[\bar{v}]_2$, whose exponent corresponds to evaluating the "target monomial" $\prod_{k \in \mathbb{Z}_w} X_k$ at $\mathbf{v}$. Notice that the target monomial is set up in such a way that $[\bar{v}]_t = [v_i]_1 \cdot [\bar{v}_i]_2$ holds for all $i \in \mathbb{Z}_w$, where $[\bar{v}_i]_2$ can be viewed as a "complement" of $[v_i]_1$. Consequently, the value $y = \langle \mathbf{f}, \mathbf{x} \rangle$ appears as the coefficient of $[\bar{v}]_t$ in the inner product $\left\langle \sum_{i \in \mathbb{Z}_w} x_i \cdot [v_i]_1, \sum_{j \in \mathbb{Z}_w} f_j \cdot [\bar{v}_j]_2 \right\rangle$.

While the above pairing-based scheme is ready to be translated to the lattice setting using our translation strategy, to prepare for our generalised scheme for higher-degree polynomials, we divide the target and complement monomials by $\prod_{k \in \mathbb{Z}_w} X_k$. The complement of X_i becomes X_i^{-1} and the target monomial becomes the constant 1. Concretely, we divide the opening and the verification equation by $\bar{v}$ to obtain

$$[u']_2 \coloneqq \sum_{i,j \in \mathbb{Z}_w : i \neq j} f_j \cdot x_i \cdot [v_i / v_j]_2$$

$$\langle [1]_1, [u']_2 \rangle \stackrel{?}{=} \left\langle [c]_1, \sum_{j \in \mathbb{Z}_w} f_j \cdot [v_j^{-1}]_2 \right\rangle - y \cdot [1]_t.$$

Recall that in the VC construction above we relied on the hardness of computing $[\bar{v}]_2$. What we have done here might seem absurd, since the element $[1]_2$ now is given in the group setting, but finding a short pre-image of a fixed image, say 1, is seemingly hard in the lattice setting. Indeed, translating the modified scheme, we derive the following lattice-based scheme.

- Public Parameters: $\left(\mathbf{a}, \mathbf{v}, (\mathbf{u}_{i,j})_{i \neq j \in \mathbb{Z}_w}\right)$ where $\langle \mathbf{a}, \mathbf{u}_{i,j} \rangle \equiv v_i / v_j \bmod q$, $\mathbf{u}_{i,j}$ are short
- Committing $\mathbf{x} \in \mathcal{R}^w$: $c \coloneqq \langle \mathbf{v}, \mathbf{x} \rangle \bmod q$
- Opening f: $\mathbf{u} \coloneqq \sum_{i,j \in \mathbb{Z}_w : i \neq j} f_j \cdot x_i \cdot \mathbf{u}_{i,j}$
- Verifying (f, y): $\langle \mathbf{a}, \mathbf{u} \rangle \stackrel{?}{\equiv} \left(\sum_{j \in \mathbb{Z}_w} f_j \cdot v_j^{-1} \right) \cdot c - y \bmod q$ and $\mathbf{u}$ is short

For correctness, we require that the committed vector $\mathbf{x}$ and the function f both have short coefficients.

The weak binding property of the translated lattice-based scheme relies on the hardness of finding a short preimage of (a small multiple of) 1 given short preimages of v_i / v_j for all $i, j \in \mathbb{Z}_w$ with $i \neq j$ – a new computational assumption obtained by translating its

pairing-counterpart, which belongs to a new family of assumptions called the k-R-ISIS assumption family.

Furthermore, the computation of $\sum_{j\in\mathbb{Z}_w} f_j \cdot v_j^{-1}$ in the verification equation can be preprocessed before knowing the commitment c and the opening proof $\mathbf{u}$, such that the online verification can be performed in time sublinear in w.

Supporting Higher-Degree Polynomials. Notice that in the group setting the (modified) verification algorithm can be seen as evaluating the linear form f at $([v_0^{-1}]_2 \cdot [c]_1, \ldots, [v_{w-1}^{-1}]_2 \cdot [c]_1)$ where $[c]_1$ supposedly encodes $\mathbf{x}$. In the group setting, f has to be linear since we cannot multiply two $\mathbb{G}_1$ elements together to get an encoding of the Kronecker product $\mathbf{x} \otimes \mathbf{x}$.

In the lattice setting, however, the commitment c is a ring element and thus we can evaluate a non-linear polynomial f at $(v_0^{-1} \cdot c, \ldots, v_{w-1}^{-1} \cdot c)$. Moreover, we notice that each degree-d monomial $\mathbf{x}^{\mathbf{e}}$ is encoded in c^d as (a factor of) the coefficient of $\mathbf{v}^{\mathbf{e}}$, which has a natural complement $\mathbf{v}^{-\mathbf{e}}$ satisfying $(\mathbf{v}^{\mathbf{e}}) \cdot (\mathbf{v}^{-\mathbf{e}}) = 1$, our modified target monomial. This suggests the possibility of generalising the translated lattice-based scheme above to support openings to higher-degree polynomials. Indeed, this technique allows us to generalise the scheme to support bounded-coefficient polynomials of degrees up to a constant, whose weak binding property is now based on another member of the k-R-ISIS assumption family.

Achieving Compactness and Extractability. The VC scheme obtained above achieves succinctness, i.e. commitments and opening proofs are of size sublinear in w (not t), and weak binding, which fall short of the compactness and extractability required to construct a SNARK. Indeed, a black-box construction of SNARK using this VC is unlikely since, so far, we are only relying on falsifiable assumptions. To resolve this problem, we propose a knowledge version of the k-R-ISIS assumptions. For concreteness, we will use the following member of the knowledge k-R-ISIS assumption family:

> *Let* $\mathbf{a}' \leftarrow \mathcal{R}_q^\ell$ *and* $\mathbf{v} \leftarrow \mathcal{R}_q^w$ *be random vectors and* $t \leftarrow \mathcal{R}_q$ *be a random element such that* $|t \cdot \mathcal{R}_q|$ *is super-polynomial in* λ *and* $|t \cdot \mathcal{R}_q|/|\mathcal{R}_q|$ *is negligible in* λ. *If there exists an efficient algorithm* $\mathcal{A}$ *which, given short vectors* $\mathbf{u}_i'$ *satisfying* $\langle \mathbf{a}', \mathbf{u}_i' \rangle = v_i \cdot t \bmod q$ *for all* $i \in \mathbb{Z}_w$, *produces* $(c, \mathbf{u}')$ *such that* $\mathbf{u}'$ *is a short vector satisfying* $\langle \mathbf{a}', \mathbf{u}' \rangle = c \cdot t \bmod q$, *then there exists an efficient extractor* $\mathcal{E}_{\mathcal{A}}$ *which extracts a short vector* $\mathbf{x} \in \mathcal{R}^w$ *such that* $\langle \mathbf{v}, \mathbf{x} \rangle = c \bmod q$.

Equipped with this k-R-ISIS of knowledge assumption, we can upgrade our VC construction to achieve extractability as follows. First, we let the public parameters to additionally include $(\mathbf{a}', (\mathbf{u}_i')_{i\in\mathbb{Z}_w}, t)$. Here t generates an ideal that is small enough for random elements in $\mathcal{R}_q$ not to be contained within it, but big enough to provide sufficient entropy. Next, we let the committer also include $\mathbf{u}' = \sum_{i\in\mathbb{Z}_w} x_i \cdot \mathbf{u}_i'$ in an opening proof. Finally, we let the verifier additionally check that $\mathbf{u}'$ is short and $\langle \mathbf{a}', \mathbf{u}' \rangle = c \cdot t \bmod q$.

To see why the modified scheme is extractable, suppose an adversary is able to produce a commitment c and a valid opening proof for (f, y). By the k-R-ISIS of knowledge assumption, we can extract a short vector $\mathbf{x} \in \mathcal{R}^w$ such that $\langle \mathbf{v}, \mathbf{x} \rangle = c \bmod q$. Now, if $f(\mathbf{x}) = y' \neq y$, we can use the extracted $\mathbf{x}$ to compute a valid opening proof for (f, y'). However, being able to produce valid opening proofs for both (f, y) and (f, y') with $y \neq y'$ violates the weak binding property. We therefore conclude that $f(\mathbf{x}) = y$.

It remains to show how we can achieve compactness. Since our lattice-based VC schemes preserve the property of the original pairing-based schemes that the verification algorithm is linearly-homomorphic in the opening proofs, a natural strategy towards compactness is to aggregate multiple opening proofs into one using a random linear combination, with coefficients generated using a random oracle. The binding property of an aggregated opening proof can be proven using a classic rewinding argument which involves inverting a Vandermode matrix defined by the randomness used for aggregation. This strategy works particularly well in the prime-order group setting since scalars are field elements and Vandermonde matrices defined by distinct field elements are always invertible. In the lattice setting, however, the coefficients used for aggregation have to be chosen from a set where the difference between any pair of elements is (almost) invertible (over $\mathcal{R}$) for an analogous argument to go through. This is a severe limitation since sets satisfying this property cannot be too large [AL21].

To achieve compactness in the lattice setting, we are forced to use a different strategy. Specifically, the coefficients $\mathbf{h} = (h_i)_{i \in \mathbb{Z}_t} \in \mathcal{R}$ that we use to aggregate opening proofs are given by an instance of the R-SIS problem over $\mathcal{R}_p$ (taking smallest $\mathcal{R}$-representatives of $\mathcal{R}_p$ elements) sampled as part of the public parameters, where p is chosen such that the R-SIS assumption is believed to hold over $\mathcal{R}_p$ while p is small relative to q.

To see why extractability still holds, suppose an adversary is able to produce a commitment c and a valid opening proof for (f, y) where $f = \sum_{i \in \mathbb{Z}_t} h_i \cdot f_i$ and $y = \sum_{i \in \mathbb{Z}_t} h_i \cdot y_i$. By our previous argument, we can extract $\mathbf{x}$ satisfying $f(\mathbf{x}) = y$. Suppose it is not the case that $f_i(\mathbf{x}) = y_i$ for all $i \in \mathbb{Z}_t$, then $(f_i(\mathbf{x}) - y_i)_{i \in \mathbb{Z}_t}$ is a short vector satisfying $\sum_{i \in \mathbb{Z}_t} h_i \cdot (f_i(\mathbf{x}) - y_i) = 0$ over $\mathcal{R}$, which implies $\sum_{i \in \mathbb{Z}_t} h_i \cdot (f_i(\mathbf{x}) - y_i) = 0 \bmod p$, breaking the R-SIS assumption over $\mathcal{R}_p$.

Discussion and Generalisations. We discuss the resulting VC scheme obtained through the aforementioned series of transformations. Our VC scheme supports openings to w-variate t-output constant-degree polynomial maps with bounded coefficients. The scheme achieves compactness and extractability, where the latter is based on the standard R-SIS assumption over $\mathcal{R}_p$ and our two new assumptions: k-R-ISIS and the k-R-ISIS of knowledge assumption over $\mathcal{R}_q$, where p is short relative to q. The construction uses only algebraic operations over $\mathcal{R}$ and $\mathcal{R}_q$. Furthermore, a major part of the verification equation can be precomputed, so that the online verification time is sublinear in w and t.

Our construction and the k-R-ISIS (of knowledge) assumption families admit natural generalisations to the module setting, where the vector $\mathbf{a}$ is replaced by a matrix $\mathbf{A}$ and other components are modified accordingly. Expectedly, we show that the module

versions of the k-R-ISIS assumptions are at least as hard as the ring versions for certain parameter choices.

In many applications (e.g. aggregating signatures), often only a main part (e.g. a set of signature verification keys) of the function-image tuple (f, y) is known in advance, while the remaining small part (e.g. a message signed by all parties) is known when it comes the time to perform verification. It is desirable to preprocess the main part of (f, y) offline, so that the online verification cost is only dependent on the size of the small part. In our formal construction, we capture this flexibility by considering y itself to be a polynomial map, and allowing f and y to take an (additional, for f) public input $\mathbf{z}$. This allows the maps (f, y) to be preprocessed, such that the online cost depends mostly on $\mathbf{z}$.

3.1.4 Application

We highlight an application of interest of our VC, and in particular of the resulting SNARK, in aggregating GPV signatures [GPV08]. As a bonus result, we also show how to build adaptor signatures [AEE+21] based on GPV signatures while preserving aggregatability. For more comprehensive details we refer the reader to Section 3.6.2 and Appendix B.2.

Aggregate GPV Signatures. GPV signatures [GPV08] are a lattice-based signature scheme paradigm of which an instantiation is a finalist in the NIST Post-Quantum Process (Falcon [PFH+20]). On a high level, a GPV signature on a message m is a short vector $\mathbf{u}$ such that $\mathbf{A} \cdot \mathbf{u} \equiv \mathbf{v} \bmod q$, where $\mathbf{A}$ is the public key, $\mathbf{v} = H(m)$ with the hash function H modelled as a random oracle in the security analysis. The verification is simply the check of the linear relation $\mathbf{A} \cdot \mathbf{u} \equiv \mathbf{v} \bmod q$ and that $\mathbf{u}$ is short.

Our SNARK can be used to prove knowledge of GPV signatures natively given the signature verification involves algebraic operations only. For instance, to aggregate n signatures $(\mathbf{u}_i)_{i \in \mathbb{Z}_n}$ on the same message m (a scenario that arises in a PoS consensus protocol [DGNW20]), the aggregator can compute a SNARK proof of knowledge of short $(\mathbf{u}_i)_{i \in \mathbb{Z}_n}$ satisfying $\mathbf{A}_i \cdot \mathbf{u}_i = \mathbf{v} \bmod q$, where $\mathbf{A}_i$ is the public key of the i-th signer. The aggregated signature i.e. the SNARK proof, can be verified in time sublinear in the number of signers and signatures n by first preprocessing the part of the verification equation depending on $(\mathbf{A}_i)_{i \in \mathbb{Z}_n}$. In fact, this preprocessing step is one-time for the given set of signers, and the online verification after knowing m is only logarithmic in n. If the signers sign different messages, a similar SNARK but now over the different messages results in a compact proof, but with verification time linear in n (similar to the case of BLS signatures [BDN18]). Such aggregation can result in compact blocks in a blockchain as shown for the case of BLS signatures [BDN18], but now with post-quantum security.

Aggregate Adaptor Signatures. Adaptor signatures [AEE+21, EEE20, AME+21] let a user generate an encryption $\hat{\sigma}$ of a signature σ on a message m with respect to an instance Y of a hard language $\mathcal{L}$. Here $\hat{\sigma}$ is also referred to as a *pre-signature*. Given the public key, it is efficient to verify if a given pre-signature $\hat{\sigma}$ is indeed valid with respect to

the instance and the message. One can *adapt* the pre-signature $\hat{\sigma}$ into a valid signature σ given the witness y for the instance Y, and given $\hat{\sigma}$ and σ one can efficiently *extract* the witness y. The primitive has found itself useful in enhancing efficiency and privacy of conditional payments in cryptocurrencies [AEE+21, AME+21], and aggregation of signatures adds clear benefits to this primitive. In the following we discuss how GPV signatures can be turned into adaptor signatures, which consequently implies that they can be aggregated via our newly constructed SNARK.

We consider the lattice trapdoor from [MP12] for our GPV signatures, and view the GPV signatures as follows. The public parameters are given by a uniformly random matrix $\mathbf{A}$, the signing key is $\mathsf{sk} := \mathbf{X}$, where $\mathbf{X}$ is a short norm matrix such that the public key, $\mathbf{Y} := \mathbf{A} \cdot \mathbf{X} \bmod q$, is distributed statistically close to random. The signature is simply $(\mathbf{z}, \mathbf{c})$ such that during verification we have $[\mathbf{A}|\mathbf{G}+\mathbf{Y}] \cdot [\mathbf{z}|\mathbf{c}]^{\mathsf{T}} = H(m) \bmod q$ and $\|(\mathbf{c}, \mathbf{z})\|$ is small as stipulated by GPV signatures. Here $\mathbf{G}$ is the gadget matrix. We choose the hard language

$$\mathcal{L} := \{(\mathbf{A}, \mathbf{v}') : \exists\ \mathbf{u}'\ s.t.\ \mathbf{A} \cdot \mathbf{u}' = \mathbf{v}' \wedge \|\mathbf{u}'\| \leq \beta^*\},$$

where $\mathbf{A} \in \mathcal{R}_q^{\eta \times \ell}$, $\mathbf{v}' \in \mathcal{R}_q^{\eta}$. A pre-signature $\hat{\sigma}$ is simply $(\mathbf{c}, \hat{\mathbf{z}})$ with $\mathbf{v}'$ as the hard instance, such that during pre-signature verification, it holds that $[\mathbf{A}|\mathbf{G}+\mathbf{Y}] \cdot [\hat{\mathbf{z}}|\mathbf{c}]^{\mathsf{T}} = H(m) - \mathbf{v}' \bmod q$ and $\|(\mathbf{c}, \hat{\mathbf{z}})\|$ is small. It is easy to adapt $\hat{\sigma}$ given the witness $\mathbf{u}'$ by setting $\mathbf{z} := \hat{\mathbf{z}} + \mathbf{u}'$ and $\sigma := (\mathbf{c}, \mathbf{z})$. To extract a witness one can simply compute $\mathbf{u}' := \mathbf{z} - \mathbf{z}'$. Similar to [EEE20] we have that the extracted $\mathbf{u}'$ has a slightly higher norm than that was used to adapt the pre-signature. The security of our scheme only relies on the M-SIS problem and the RO model.

3.1.5 Related Work

Apart from applications to succinct arguments [LM19], VCs have found numerous applications, such as verifiable databases [CF13], verifiable decentralized storage [CFG+20], updatable zero-knowledge sets [MRK03, Lis05], keyless Proofs of Retrievability (PoR) [Fis18, Fis19], pseudonymous credentials [KZG10], and cryptocurrencies with stateless transaction validation [CPZ18]. Several works have studied various extensions to VC, with updatable commitments and proofs [CF13], aggregatable opening proofs for different commitments [GRWZ20], and incremental aggregatable proofs [CFG+20].

Libert, Ramanna, and Yung [LRY16] showed that a VC for linear functions over $\mathbb{Z}_q$ implies a polynomial commitment for polynomials over $\mathbb{Z}_q$. The result was obtained by VC-committing to the coefficient vector of the polynomial and opening it to a linear function whose coefficients are evaluations of monomials at the evaluation point. Since our VC only allows committing to a short vector $\mathbf{x} \in \mathcal{R}^w$ and opening to a polynomial map f with short coefficients, we need to suitably tune the norm bound α of f and $\mathbf{x}$ to obtain similar applications. Concretely, by setting $\alpha \approx \delta^{d+1} \cdot \gamma_{\mathcal{R}}^d$ where $\gamma_{\mathcal{R}}$ is the ring expansion factor of $\mathcal{R}$, we obtain a polynomial commitment for degree-d multivariate polynomials with coefficients bounded by δ which supports evaluations at vectors of norm

also bounded by δ. Note that only constant-degree polynomials are supported by our polynomial commitment since α depends exponentially on d.

In the same work [LRY16], Libert, Ramanna, and Yung also showed that the polynomial commitment constructed from a VC for linear functions over $\mathbb{Z}_q$ implies an accumulator for $\mathbb{Z}_q$ elements, the construction requires committing to the polynomial $p(X) = \prod_{a \in A}(X - a)$ encoding the set A of elements to be accumulated. The polynomial commitment obtained via our VC unfortunately does not support committing to $p(X)$ since its degree is as large as $|A|$.

In a recent work [PPS21], Peikert, Pepin, and Sharp proposed a VC for positional openings based on the standard SIS assumption. Relative to our construction outlined in Section 3.1.3, their construction can be interpreted as follows. Instead of handing out preimages $\mathbf{u}_{i,j}$ with $\langle \mathbf{a}, \mathbf{u}_{i,j} \rangle = v_j / v_i \bmod q$, they sample multiple $\mathbf{a}_i$ for $i \in \mathbb{Z}_w$ and let $\mathbf{u}_{i,j}$ satisfy $\langle \mathbf{a}_i, \mathbf{u}_{i,j} \rangle = v_j \bmod q$. To verify an opening to position i, the vector $\mathbf{a}_i$ is used. The removal of the non-linear term v_j / v_i allows proving security from the SIS assumption. On the flip side, using a different vector $\mathbf{a}_i$ to verify openings to different positions i forbids the standard technique of aggregating openings using a random linear combination. Furthermore, there seems to be no natural way of generalising their construction to support functional openings without significantly changing the VC model, e.g. introducing an authority responsible for issuing functional opening keys [PPS21]. Even if we consider the model with an authority, the resulting VC only satisfies *weak binding* (using the terminology of our work) making it unsuitable to be transformed into a SNARG: There is in fact an explicit attack when compiling their VC (with authority) into a SNARG.[4]

In another recent work [AKSY21] Agrawal, Kirshanova, Stehlé, and Yadav constructed a blind signature scheme from a novel SIS-like assumption of the "one-more" flavour. Here the adversary can query ℓ arbitrary preimages for an ISIS instance and must then output $\ell + 1$ preimages of random images returned by an oracle. While this assumption is in the same "spirit" as those introduced in this work, they seem incomparable: being adaptive makes one-more-SIS potentially easier, requiring preimages of random images (hence without structure) seems to make it harder.

Prior to our work, all lattice-based SNARKs were in the designated-verifier setting. These constructions [GMNO18, ISW21] are based on "linear-only" assumptions which are similar in spirit to the knowledge k-M-ISIS assumptions introduced in this work but with a key difference: While linear-only assumptions are with respect to specific encryption schemes, our assumptions are with respect to general rings. In terms of applications, linear-only encryption has always been used to construct designated-verifier primitives. In contrast, knowledge k-M-ISIS naturally leads to constructions of publicly verifiable primitives.

[4] We stress that this does not contradict any of the claims made in [PPS21], but rather exemplifies the difference between their approach and ours.

3.2 Preliminaries

Let $\lambda \in \mathbb{N}$ denote the security parameter. Define $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Let $\mathcal{R}$ be a ring. We write $\mathcal{R}[\mathbf{X}]$ for the (multivariate) polynomial ring over $\mathcal{R}$ and $\mathcal{R}(\mathbf{X})$ for the ring of (multivariate) rational functions over $\mathcal{R}$ with intermediates $\mathbf{X} = (X_i : i \in \mathbb{Z}_w)$. We write $\langle \mathcal{G} \rangle$ for the ideal resp. module spanned by the elements of the set $\mathcal{G} \subset \mathcal{R}^\eta$ for $\eta \in \mathbb{N}$. When $\mathcal{G}$ is a singleton set we may suppress the $\{\cdot\}$ notation. We write $|\langle \mathcal{G} \rangle|$ for size of the ideal $\langle \mathcal{G} \rangle$ as a set.

For $m \in \mathbb{N}$, let $\zeta_m \in \mathbb{C}$ be any fixed primitive m-th root of unity. Denote by $\mathcal{K} = \mathbb{Q}(\zeta_m)$ the cyclotomic field of order $m \geq 2$ and degree $n = \varphi(m)$, and by $\mathcal{R} = \mathbb{Z}[\zeta_m]$ its ring of integers, called a cyclotomic ring for short. We have $\mathcal{R} \cong \mathbb{Z}[x]/\langle \Phi_m(x) \rangle$, where $\Phi_m(x)$ is the m-th cyclotomic polynomial. If m is a power of 2, we call $\mathcal{R}$ a power-of-2 cyclotomic ring. If m is a prime-power, we call $\mathcal{R}$ a prime-power cyclotomic ring. Let $q \in \mathbb{N}$ be prime, we write $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$ and $\mathcal{R}_q^\times$ for all invertible elements in $\mathcal{R}_q$. We have that $\mathcal{R}_q$ splits into f fields of degree $\phi(m)/f$. We write $\mathrm{vec}(r) \in \mathbb{Z}^n$ for the coefficient vector of r (with the powerful basis). For any $r \in \mathcal{R}$ there exists a matrix $\mathrm{rot}(r) \in \mathbb{Z}^{n \times n}$ s.t. $\forall s \in \mathcal{R}$ we have $\mathrm{vec}(r \cdot s) = \mathrm{rot}(r) \cdot \mathrm{vec}(s)$. For elements $x \in \mathcal{R}$ we denote the infinity norm of its coefficient vector as $\|x\| := \|\mathrm{vec}(x)\|$. If $\mathbf{x} \in \mathcal{R}^\ell$ we write $\|\mathbf{x}\|$ for the infinity norm of $\mathbf{x}$. We write $\|\cdot\|_p$ for the ℓ_p-norm, e.g. $\|\cdot\|_2$ for the Euclidean norm. We write $\mathcal{M}_{\mathcal{G}}(\cdot)$ for a function that takes vectors indexed by $\mathcal{G}$ and returns a matrix where each column corresponds to one such vector. We write $\mathbf{I}_n$ for the identity matrix of dimension n over whatever ring is clear from context.

For $w \in \mathbb{N}$, $\mathbf{x} = (x_i : i \in \mathbb{Z}_w) \in \mathcal{R}^w$, and $\mathbf{e} = (e_i : i \in \mathbb{Z}_w) \in \mathbb{Z}^w$, we write $\mathbf{x}^\mathbf{e} := \prod_{i \in \mathbb{Z}_w} x_i^{e_i}$ whenever it is defined. For $\mathbf{v} = (v_i : i \in \mathbb{Z}_w) \in (\mathcal{R}_q^\times)^w$, we write $\bar{\mathbf{v}} := (v_i^{-1} : i \in \mathbb{Z}_w)$ for the entry-wise inverse of $\mathbf{v}$. A Laurent monomial $g(\mathbf{X}) \in \mathcal{R}(\mathbf{X})$ is an expression $g(\mathbf{X}) = \mathbf{X}^\mathbf{e} := \prod_{i \in \mathbb{Z}_w} X_i^{e_i}$ with exponent vector $\mathbf{e} = (e_i : i \in \mathbb{Z}_w) \in \mathbb{Z}^w$.

We may suppress arbitrary subscripts and superscripts from problem and advantage notations when those are clear from context. We write $x \leftarrow \mathcal{D}$ for sampling from the distribution $\mathcal{D}$ and $x \leftarrow\!\!\$ \mathcal{S}$ to sample an element from the finite space $\mathcal{S}$ uniformly at random. We write $U(\mathcal{S})$ for the uniform distribution over $\mathcal{S}$ and $\{\mathbf{u}_\mathcal{G}\} := \{\mathbf{u}_g\}_{g \in \mathcal{G}}$.

Definition 3.2.1 (Ring Expansion Factor)**.** *Let $\mathcal{R}$ be a ring. The expansion factor of $\mathcal{R}$, denoted by $\gamma_\mathcal{R}$, is $\gamma_\mathcal{R} := \max_{a,b \in \mathcal{R}} \frac{\|a \cdot b\|}{\|a\| \cdot \|b\|}$.*

Proposition 1 ([AL21])**.** *If $\mathcal{R} = \mathbb{Z}[\zeta_m]$ is a prime-power cyclotomic ring, then $\gamma_\mathcal{R} \leq 2\,n$. If $\mathcal{R} = \mathbb{Z}[\zeta_m]$ is a power-of-2 cyclotomic ring, then $\gamma_\mathcal{R} \leq n$.*

Proposition 2. *Let $q = \omega((w \cdot f)^{f/\phi(m)})$ be a rational prime such that $\mathcal{R}_q$ splits into f fields each of size $q^{\varphi(m)/f}$. For $\mathbf{v} \leftarrow\!\!\$ \mathcal{R}_q^w$, we have $\mathbf{v} \in (\mathcal{R}_q^\times)^w$ with non-negligible probability.*

Proof. The probability that $\mathbf{v} \in (\mathcal{R}_q^\times)^w$ is $(1 - 1/q^{\varphi(m)/f})^{w \cdot f} \geq 1 - (w \cdot f)/q^{\varphi(m)/f}$ which is non-negligible. □

For the rest of this work, we implicitly assume q is large enough so that a uniformly random $\mathbf{v} \leftarrow \mathcal{R}_q^w$ satisfies $\mathbf{v} \in (\mathcal{R}_q^\times)^w$ with non-negligible probability.

3.2.1 Lattices

We write $\Lambda(\mathbf{B})$ for the Euclidean lattice generated by the columns of $\mathbf{B} \in \mathbb{Z}^{n \times d} = [\mathbf{b}_0 \dots \mathbf{b}_{d-1}]$, i.e. $\{ z_i \cdot \mathbf{b}_i \mid z_i \in \mathbb{Z} \}$. When $\mathbf{B}$ has full rank we call it a basis and when $n = d$ we say that $\Lambda(\mathbf{B})$ has full rank. The determinant of a full rank lattice is the absolute value of the determinant of any of its bases. Minkowski's theorem implies that there is a vector $\mathbf{x} \in \Lambda \subset \mathbb{R}^d$ of (infinity) norm $\|\mathbf{x}\| \leq \det(\Lambda)^{1/d}$ when Λ has full rank. The Gaussian heuristic predicts that a random full-rank lattice Λ contains a shortest vector of (Euclidean) norm $\approx \sqrt{\frac{d}{2\pi e}} \cdot \det(\Lambda)^{1/d}$.

For any $\mathbf{c} \in \mathbb{R}^n$ and any real $\sigma > 0$, the (spherical) Gaussian function with standard deviation parameter σ and centre $\mathbf{c}$ is:

$$\forall \mathbf{x} \in \mathbb{R}^n, \rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp\left(-\frac{\pi \cdot \|\mathbf{x} - \mathbf{c}\|_2^2}{\sigma^2}\right).$$

The Gaussian distribution is $\mathcal{D}_{\sigma,\mathbf{c}}(\mathbf{x}) = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\sigma^n$. The (spherical) discrete Gaussian distribution over a lattice $\Lambda \in \mathbb{R}^n$, with standard deviation parameter $\sigma > 0$ and centre $\mathbf{c}$ is:

$$\forall \mathbf{x} \in \Lambda, \mathcal{D}_{\Lambda,\sigma,\mathbf{c}} = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(\Lambda)},$$

where $\rho_{\sigma,\mathbf{c}}(\Lambda) \coloneqq \sum_{\mathbf{x} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. When $\mathbf{c} = \mathbf{0}$ we omit the subscript $\mathbf{c}$. We may write $\mathcal{D}_{\mathcal{R},\sigma}$ where we interpret $\mathcal{R}$ to be the lattice spanned by $\mathcal{R}$.

The dual of a lattice Λ is defined by $\Lambda^* = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^\mathsf{T} \cdot \Lambda \subseteq \mathbb{Z}\}$. The smoothing parameter of an n-dimensional lattice Λ with respect to $\epsilon > 0$, denoted $\eta_\epsilon(\Lambda)$, is the smallest $\sigma > 0$, such that $\rho_{1/\sigma}(\Lambda^* \setminus \{0\}) \leq \epsilon$.

Lattice reduction with parameter κ returns a vector of Euclidean norm $\approx \delta^{d-1} \cdot \det(\Lambda)^{1/d}$ where δ is the root Hermite factor δ and a function of κ.[5] A root Hermite factor $\delta \approx (\frac{\kappa}{2\pi e})^{1/(2\kappa)}$ can be achieved in time $2^{0.292\,\kappa + o(\kappa)}$ classically using the BKZ algorithm [SE94] with block size κ and sieving as the SVP oracle [BDGL16] (quantum algorithms do not promise a sufficiently substantial speed-up [Laa15, AGPS20]). Concretely, for $\lambda = 128$ we require $\kappa \geq 484$ and thus $\delta \leq 1.0034$.

3.2.2 Sampling Algorithms

The following relies on analogues of the Leftover Hash Lemma over rings attesting that given $\mathbf{a}_i \leftarrow U(\mathcal{R}_q^\eta)$ and $r_i \leftarrow \mathcal{D}$ where $\mathcal{D}$ is a small uniform [Mic07, SSTX09] or discrete Gaussian distribution [SS11, LPR13], we have that $\left(\mathbf{a}_0, \dots, \mathbf{a}_{\ell-1}, \sum_{0 \leq i < \ell} \mathbf{a}_i \cdot r_i\right)$ is close

[5] The literature routinely simplifies the first expression to $\approx \delta^d \cdot \det(\Lambda)^{1/d}$

to uniform. In what follows, we will write $\mathsf{lhl}(\mathcal{R}, \eta, q, \mathcal{D})$ for an algorithm that outputs a minimal $\ell \in \mathbb{N}$ ensuring that the resulting distribution is within negl to uniform. We may also write $\mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$ for some $\mathcal{D}$ outputting elements bounded by β (with overwhelming probability). In many cases the reader may think $\ell \in O(\eta \log_\beta(q))$. Let $(\mathsf{TrapGen}, \mathsf{SampD}, \mathsf{SampPre})$ be PPT algorithms with the following syntax and properties [GPV08, MP12, GM18]:

- $(\mathbf{A}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\eta, 1^\ell, q, \mathcal{R}, \beta)$ takes dimensions $\eta, \ell \in \mathbb{N}$, a modulus $q \in \mathbb{N}$, a ring $\mathcal{R}$, and a norm bound $\beta \in \mathbb{R}$. It generates a matrix $\mathbf{A} \in \mathcal{R}_q^{\eta \times \ell}$ and a trapdoor td. For any $n \in \mathsf{poly}(\lambda)$ and $\ell \geq \mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$, the distribution of $\mathbf{A}$ is within negl statistical distance of $U(\mathcal{R}_q^{\eta \times \ell})$.
- $\mathbf{u} \leftarrow \mathsf{SampD}(1^\eta, 1^\ell, \mathcal{R}, \beta')$ with $\ell \geq \mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$ outputs an element in $\mathbf{u} \in \mathcal{R}^\ell$ with norm bound $\beta' \geq \beta$. We have that $\mathbf{v} \coloneqq \mathbf{A} \cdot \mathbf{u} \bmod q$ is within negl statistical distance to $U(\mathcal{R}_q^\eta)$.
- $\mathbf{u} \leftarrow \mathsf{SampPre}(\mathsf{td}, \mathbf{v}, \beta')$ with $\ell \geq \mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$ takes a trapdoor td, a vector $\mathbf{v} \in \mathcal{R}_q^\eta$, and a norm bound $\beta' \geq \beta$. It samples $\mathbf{u} \in \mathcal{R}^\ell$ satisfying $\mathbf{A} \cdot \mathbf{u} \equiv \mathbf{v} \bmod q$ and $\|\mathbf{u}\| \leq \beta'$. Furthermore, $\mathbf{u}$ is within negl statistical distance to $\mathbf{u} \leftarrow \mathsf{SampD}(1^\eta, 1^\ell, \mathcal{R}, \beta')$ conditioned on $\mathbf{v} \equiv \mathbf{A} \cdot \mathbf{u} \bmod q$. The syntax can be extended in the natural way for $\mathsf{SampPre}$ to take a matrix $\mathbf{V}$ as input, in which case $\mathsf{SampPre}$ is run on each column of $\mathbf{V}$ and the output vectors are concatenated column-wise to form a matrix.

For all algorithms we may replace β by $\mathcal{D}$ where it is understood that $\mathcal{D}$ outputs samples bounded by β (with overwhelming probability).

Proposition 3 (adapted from Lemma 5 of [AKSY21])**.** *For any $k > 1/\sqrt{2\,\pi}$,*

$$\Pr \|\mathbf{z}\|_2 > k \cdot \sigma \cdot \sqrt{2\pi\, n};\ \mathbf{z} \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sigma} < \left(k \cdot \sqrt{2\pi}\right)^n \exp\left(\frac{n}{2} \cdot \left(1 - 2\pi\, k^2\right)\right),$$
$$\Pr \|\mathbf{z}\|_\infty > k \cdot \sigma;\ \mathbf{z} \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sigma} < 2\, n \cdot \exp(-\pi k^2).$$

3.2.3 Rényi Divergence

Definition 3.2.2. *Let P and Q be any two discrete probability distributions such that $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$. Then for $a \in (1, \infty)$, the Rényi Divergence (RD) of order a is defined by*

$$R_a(P \| Q) = \left(\sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{\frac{1}{a-1}}.$$

Lemma 3.2.1 (in Lemma 2.9 of [BLR+18])**.** *Let P and Q be any two discrete probability distributions such that $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$ and let $a \in (1, \infty)$.*

- *Let $E \subseteq \mathrm{Supp}(Q)$ be an arbitrary event, then $Q(E) \geq P(E)^{\frac{a}{a-1}} / R_a(P \| Q)$.*

- *Assume P and Q are two distributions of a pair of random variable (Y_0, Y_1). For $i \in \{0,1\}$ let P_i (resp. Q_i) denote the marginal distribution of Y_i under P (resp. Q), and let $P_{1|0}(\cdot|y_0)$ (resp. $Q_{1|0}(\cdot|y_0)$) denote the conditional distribution of Y_1 given that $Y_0 = y$. Then we have*

$$R_a(P||Q) = R_a(P_0||Q_0) \cdot R_a(P_1||Q_1) \text{ if } Y_0 \text{ and } Y_1 \text{ are independent.}$$

Lemma 3.2.2 ([BLR+18])**.** *For any n-dimensional lattice, $\Lambda \in \mathbb{R}^n$ and $\sigma > 0$, let P be the distribution $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$, and Q be the distribution $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}'}$ for some fixed $\mathbf{c}, \mathbf{c}' \in \mathbb{R}^n$. If $\mathbf{c}, \mathbf{c}' \in \Lambda$, let $\epsilon = 0$. Otherwise fix $\epsilon \in (0,1)$ and assume that $\sigma > \eta_\epsilon(\Lambda)$. Then for any $a \in (1, \infty)$*

$$R_a(P\|Q) \in \left[\left(\frac{1-\epsilon}{1+\epsilon}\right)^{\frac{2}{a-1}}, \left(\frac{1+\epsilon}{1-\epsilon}\right)^{\frac{2}{a-1}}\right] \cdot \exp\left(a \cdot \pi \frac{\|\mathbf{c}-\mathbf{c}'\|_2^2}{\sigma^2}\right).$$

3.2.4 Hard Problems

The Short Integer Solution problem was introduced in the seminal work of Ajtai [Ajt96]. It asks to find a short element (of Euclidean norm β_2) in the kernel of a random matrix mod q. An inhomogeneous version, asking to find a short solution to a linear algebra problem mod q was formalised later [Mic07].

For both problems, it was shown [GPV08] that solving the problem for $q \geq \beta_2 \cdot \omega(\sqrt{n \cdot \log n})$ implies solving certain presumed hard lattice problems (finding a short basis) to within approximation factor $\beta_2 \cdot \tilde{\mathcal{O}}(\sqrt{n})$. Thus, since $\beta_2 \geq \beta_\infty$, an appropriate choice of parameters is $n = \mathsf{poly}(\lambda)$, $q \geq \beta_\infty \cdot n \cdot \log n$ and $\ell \geq 2\, n \log_{\beta_\infty} q$. An algorithm solving ISIS can be used to solve SIS (by making one of the columns of $\mathbf{A}$ the target) and solving ISIS twice allows to solve SIS by considering the difference of these solutions. Ring variants were introduced in [Mic07, PR06, LM06]; module variants in [LS15].

Definition 3.2.3 (M-SIS, adapted from [LS15])**.** *Let $\mathcal{R}, \eta, q, \ell, \beta$ depend on λ. The Module-SIS (or M-SIS) problem, denoted M-$\mathsf{SIS}_{\mathcal{R}_q,\eta,\ell,\beta^*}$, is: Given a uniform $\mathbf{A} \leftarrow \mathcal{R}_q^{\eta \times \ell}$, $\mathbf{t} \equiv 0 \bmod q$ find some $\mathbf{u} \neq \mathbf{0} \in \mathcal{R}^\ell$ such that $\|\mathbf{u}\| \leq \beta^*$ and $\mathbf{A} \cdot \mathbf{u} \equiv \mathbf{t} \bmod q$. We write $\mathsf{Adv}^{\text{m-sis}}_{\mathcal{R}_q,\eta,\ell,\beta^*,\mathcal{A}}$ for the advantage of any algorithm $\mathcal{A}$ in solving M-$\mathsf{SIS}_{\mathcal{R}_q,\eta,\ell,\beta^*}$. We assume $\mathsf{Adv}^{\text{m-sis}}_{\mathcal{R}_q,\eta,\ell,\beta^*,\mathcal{A}} \leq \mathsf{negl}(\lambda)$ for appropriately chosen $\mathcal{R}_q, \eta, \ell, \beta^*$ and PPT $\mathcal{A}$. When $\mathbf{t} \neq 0$ we speak of the Module-ISIS or M-ISIS problem, denoted M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta,\ell,\beta^*}$. When $\eta = 1$ we speak of Ring-(I)SIS or R-(I)SIS, denoted R-$\mathsf{SIS}_{\mathcal{R}_q,\ell,\beta^*}$ or R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,\beta^*}$.*

In [LS15] it was shown that solving Module-SIS is as hard as finding a short basis in modules. In [LM06, PR06] it was shown that solving Ring-SIS is as hard as find a short vector in any ideal in $\mathcal{R}$. A similar result was established for Ring-ISIS [Mic07]. From a cryptanalytic perspective, no known algorithm solves Ring/Module-(I)SIS significantly faster than those solving (I)SIS. Our assumption is a generalisation and adaptation to more general rings of the k-SIS assumption.

Definition 3.2.4 (k-M-SIS, generalised from [BF11, LPSS14])**.** *For any integer $k \geq 0$, an instance of the k-M-$\mathsf{SIS}_{\mathcal{R}_q,\eta,\ell,\beta,\beta^*}$ problem is a matrix $\mathbf{A} \in \mathcal{R}_q^{\eta\times\ell}$ and a set of k vectors $\mathbf{u}_0, \dots \mathbf{u}_{k-1}$ s.t. $\mathbf{A} \cdot \mathbf{u}_i \equiv \mathbf{0} \bmod q$ with $\|\mathbf{u}_i\| \leq \beta$. A solution to the problem is a nonzero vector $\mathbf{u} \in \mathcal{R}^\ell$ such that*

$$\|\mathbf{u}\| \leq \beta^*, \quad \mathbf{A} \cdot \mathbf{u} \equiv \mathbf{0} \bmod q, \quad \text{and} \quad \mathbf{u} \notin \mathcal{K}\text{-span}(\{\, \mathbf{u}_i \,\}_{0\leq i<k}).$$

If $\mathcal{B}$ is an algorithms that takes as input a matrix $\mathbf{A} \in \mathcal{R}_q^{\eta\times\ell}$ and vectors $\mathbf{u}_i \in \mathcal{R}^\ell$ for $0 \leq i < k$, we define $\mathsf{Adv}^{\mathsf{k\text{-}m\text{-}sis}}_{\mathcal{R}_q,\eta,\ell,\beta,\beta^*,\mathcal{B}}$ to be the probability that $\mathcal{B}$ outputs a solution to the k-M-$\mathsf{SIS}_{\mathcal{R}_q,\eta,\ell,\beta,\beta^*}$ problem instance $\mathbf{A}, \mathbf{u}_0, \dots, \mathbf{u}_{k-1}$ over uniformly random $\mathbf{A} \in \mathcal{R}_q^{\eta\times\ell}$ and $\mathbf{u}_i$ drawn from $\mathsf{SampD}(1^\eta, 1^\ell, \mathcal{R}, \beta)$ conditioned on $\mathbf{A} \cdot \mathbf{u}_i \equiv \mathbf{0} \bmod q$.

In [BF11, LPSS14] it is shown that if SIS is hard for $\mathbb{Z}_q^{n\times(\ell-k)}$ and norm bound β then k-M-$\mathsf{SIS}_{\mathbb{Z}_q,n,\ell,\beta',\beta''}$ is hard for any $k < \ell$, and certain $\beta', \beta'' \in \mathsf{poly}(\beta)$. Looking ahead, here we are interested in k-R-$\mathsf{SIS}_{\mathcal{R}_q,\ell,\beta,\beta^*} := k$-$M$-$\mathsf{SIS}_{\mathcal{R}_q,1,\ell,\beta,\beta^*}$.

3.2.5 Vector Commitments

We define a non-interactive variant of vector commitments with preprocessing.

Definition 3.2.5 (Vector Commitments (VC))**.** *A (preprocessing non-interactive) vector commitment (VC) scheme is parameterised by the families*

$$\mathcal{F} = \{\, \mathcal{F}_{s,w,t} \subseteq \{\, f : \mathcal{R}^s \times \mathcal{R}^w \to \mathcal{R}^t \,\} \,\}_{s,w,t\in\mathbb{N}} \quad \text{and}$$
$$\mathcal{Y} = \{\, \mathcal{Y}_{s,t} \subseteq \{\, y : \mathcal{R}^s \to \mathcal{R}^t \,\} \,\}_{s,t\in\mathbb{N}}$$

of functions over $\mathcal{R}$ and an input alphabet $\mathcal{X} \subseteq \mathcal{R}$. The parameters s, w, and t are the dimensions of public inputs, secret inputs, and outputs of f respectively. The VC scheme consists of the PPT algorithms $(\mathsf{Setup}, \mathsf{Com}, \mathsf{Open}, \mathsf{PreVerify}, \mathsf{Verify})$ defined as follows:

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$*: The setup algorithm generates the public parameters on input the security parameter $\lambda \in \mathbb{N}$ and the size parameters $s, w, t \in \mathbb{N}$.*
- $(c, \mathsf{aux}) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x})$*: The commitment algorithm generates a commitment c of a given vector $\mathbf{x} \in \mathcal{X}^w$ with some auxiliary opening information aux.*
- $\pi \leftarrow \mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$*: The opening algorithm generates a proof π for $f(\mathbf{z}, \cdot)$ for the public input $\mathbf{z} \in \mathcal{X}^s$ and function $f \in \mathcal{F}_{s,w,t}$.*
- $\mathsf{pp}_{f,y} \leftarrow \mathsf{PreVerify}(\mathsf{pp}, (f, y))$*: Given functions $f \in \mathcal{F}_{s,w,t}$ and $y \in \mathcal{Y}_{s,t}$, the verification preprocessing algorithm generates the preprocessed public parameters $\mathsf{pp}_{f,y}$ for verifying proofs for (f, y).*

- $b \leftarrow \mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, c, \pi)$: *The verification algorithm inputs a preprocessed public parameters* $\mathsf{pp}_{f,y}$*, a public input* $\mathbf{z} \in \mathcal{X}^s$*, a commitment* c*, and an opening proof* π*. It outputs a bit* b *deciding whether to accept or reject that the vector* $\mathbf{x}$ *committed in* c *satisfies* $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$*.*

Definition 3.2.6 (Correctness)**.** *A VC scheme for* $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ *is said to be correct if for any* $\lambda, s, w, t \in \mathbb{N}$*, any* $\mathsf{pp} \in \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$*, any* $(f, \mathbf{z}, \mathbf{x}, y) \in \mathcal{F}_{s,w,t} \times \mathcal{X}^s \times \mathcal{X}^w \times \mathcal{Y}_{s,t}$ *satisfying* $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$*, any* $(c, \mathsf{aux}) \in \mathsf{Com}(\mathsf{pp}, \mathbf{x})$*, any* $\pi \in \mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$*, and any* $\mathsf{pp}_{f,y} \in \mathsf{PreVerify}(\mathsf{pp}, (f, y))$*, it holds that* $\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, c, \pi) = 1$*.*

Informally, a VC scheme is extractable if, whenever an adversary $\mathcal{A}$ is able to produce a commitment c and a valid opening proof π for some $(f(\mathbf{z}, \cdot), y(\mathbf{z}))$, then it must "know" a preimage $\mathbf{x}$ which is committed in c and satisfies $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$. Clearly, an extractable VC must also be binding, i.e. it is infeasible to open a commitment c to a set $\{ (f_i(\mathbf{z}_i, \cdot), y_i(\mathbf{z}_i)) \}_i$ of inconsistant function-image tuples.

Definition 3.2.7 (Extractability)**.** *Let* $\kappa : \mathbb{N}^4 \to [0, 1]$*. A VC scheme for* $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ *is said to be* $(\kappa, \mathcal{X}^*)$*-extractable if for any PPT adversary* $\mathcal{A}$ *there exists a PPT extractor* $\mathcal{E}_\mathcal{A}$ *such that the following probability is at most* $\kappa(\lambda, s, w, t)$*:*

$$\mathsf{Pr}\left[\begin{array}{l} \big(\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, c, \pi) = 1\big) \\ \wedge\ ((f, \mathbf{z}, \mathbf{x}, y) \notin \mathcal{F}_{s,w,t} \times \mathcal{X}^s \times (\mathcal{X}^*)^w \times \mathcal{Y}_{s,t} \\ \quad \vee\ c' \neq c \vee f(\mathbf{z}, \mathbf{x}) \neq y(\mathbf{z})) \end{array} \middle| \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t) \\ (f, y, \mathbf{z}, c, \pi) \leftarrow \mathcal{A}(\mathsf{pp}; r_\mathcal{A}) \\ (\mathbf{x}, r) \leftarrow \mathcal{E}_\mathcal{A}(\mathsf{pp}; r_\mathcal{A}) \\ (c', \mathsf{aux}') \leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x}; r) \\ \mathsf{pp}_{f,y} \leftarrow \mathsf{PreVerify}(\mathsf{pp}, (f, y)) \end{array}\right].$$

In case Com *is deterministic, we suppress the output* r *of* $\mathcal{E}_\mathcal{A}$*. We say that the scheme is* $\mathcal{X}^*$*-extractable if it is* $(\kappa, \mathcal{X}^*)$*-extractable and* $\kappa(\lambda, s, w, t)$ *is negligible in* λ *for any* $s, w, t \in \mathsf{poly}(\lambda)$*.*

Definition 3.2.8 (Compactness)**.** *A VC scheme for* $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ *is said to be compact if there exists* $p(\lambda, s, w, t) \in \mathsf{poly}(\lambda, \log s, \log w, \log t)$ *such that for any* $\lambda, s, w, t \in \mathbb{N}$*, any* $\mathsf{pp} \in \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$*, any* $(f, \mathbf{z}, \mathbf{x}, y) \in \mathcal{F}_{s,w,t} \times \mathcal{X}^s \times \mathcal{X}^w \times \mathcal{Y}_{s,t}$*, any* $(c, \mathsf{aux}) \in \mathsf{Com}(\mathsf{pp}, \mathbf{x})$*, and any* $\pi \in \mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$*, it holds that* $\max\{|c|, |\pi|\} \leq p(\lambda, s, w, t)$*, where* $|\cdot|$ *denotes the description size.*

3.2.6 SNARKs for Polynomial Map Satisfiability

We define the NP language of the satisfiability of systems of multivariate polynomials over $\mathcal{R}$ with bounded coefficients. It is straightforward to check that the language is NP-complete. In particular, it contains the NP-complete language of rank-1 constraint satisfiability (R1CS) over $\mathcal{R}$ [BCS21] as a subset.

Definition 3.2.9. *Let* $d, \alpha \in \mathbb{N}$ *with* $d \geq 2$*. The satisfiability of systems of degree-*d *polynomials over* $\mathcal{R}$ *with norm bound* α *is the language* $\mathrm{PolySAT}_{\mathcal{R},d,\alpha} = \bigcup_{s,w,t \in \mathbb{N}} \mathcal{L}_{s,w,t}$

where

$$\mathcal{L}_{s,w,t} := \{\, (f, y, \mathbf{z}) \in \mathcal{F}_{s,w,t} \times \mathcal{Y}_{s,t} \times \mathcal{X}^s : \exists\ \mathbf{x} \in \mathcal{X}^w,\ f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z}) \,\}\,.$$

where $\mathcal{F}_{s,w,t}$, $\mathcal{Y}_{s,t}$, *and* $\mathcal{X}$ *are defined as in Table 3.1.*

We recall the definition of succinct non-interactive arguments of knowledge (SNARKs). For concreteness, we state the definition with respect to the language $\mathrm{PolySAT}_{\mathcal{R},d,\alpha}$.

Definition 3.2.10 (Preprocessing Non-Interactive Arguments)**.** *A preprocessing non-interactive argument system* Π *for* $\mathrm{PolySAT}_{\mathcal{R},d,\alpha}$ *is a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{PreVerify}, \mathsf{Verify})$ *defined as follows:*

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$*: The setup algorithm generates the public parameters on input the security and size parameters* $\lambda, s, w, t \in \mathbb{N}$.
- $\pi \leftarrow \mathsf{Prove}(\mathsf{pp}, (f, y, \mathbf{z}), \mathbf{x})$*: The proving algorithm generates a proof* π *on input the public parameters* pp*, a statement* $(f, y, \mathbf{z})$*, and a witness* $\mathbf{x}$.
- $\mathsf{pp}_{f,y} \leftarrow \mathsf{PreVerify}(\mathsf{pp}, (f, y))$*: The preverification algorithm inputs the public parameters* pp *and a partial statement* (f, y)*. It outputs the preprocessed public parameters* $\mathsf{pp}_{f,y}$.
- $b \leftarrow \mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, \pi)$*: The verification algorithm returns a bit* b *(denoting acceptance or rejection) on input the preprocessed public parameters* $\mathsf{pp}_{f,y}$ *and a proof* π.

In the following definitions, we use "a system" to refer to a preprocessing non-interactive argument system for $\mathrm{PolySAT}_{\mathcal{R},d,\alpha}$.

Definition 3.2.11 (Completeness)**.** *A system* Π *is said to be complete if for any* $\lambda, s, w, t \in \mathbb{N}$*, any* $\mathsf{pp} \in \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$*, any* $(f, y, \mathbf{z}) \in \mathcal{F}_{s,w,t} \times \mathcal{Y}_{s,t} \times \mathcal{X}^s$ *and* $\mathbf{x} \in \mathcal{X}^w$ *satisfying* $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$*, any* $\pi \in \mathsf{Prove}(\mathsf{pp}, (f, y, \mathbf{z}), \mathbf{x})$*, and any* $\mathsf{pp}_{f,y} \in \mathsf{PreVerify}(\mathsf{pp}, (f, y))$*, it holds that* $\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, \pi) = 1$.

Definition 3.2.12 (Succinctness)**.** *A system* Π *is said to be succinct if for any* $\lambda, s, w, t \in \mathbb{N}$*, any* $\mathsf{pp} \in \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$*, any* $(f, y, \mathbf{z}) \in \mathcal{F}_{s,w,t} \times \mathcal{Y}_{s,t} \times \mathcal{X}^s$ *and* $\mathbf{x} \in \mathcal{X}^w$ *satisfying* $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$*, any* $\pi \in \mathsf{Prove}(\mathsf{pp}, (f, y, \mathbf{z}), \mathbf{x})$*, and any* $\mathsf{pp}_{f,y} \in \mathsf{PreVerify}(\mathsf{pp}, (f, y))$*, the runtime of* $\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, \pi)$ *is upper-bounded by a fixed polynomial in* $\mathsf{poly}(\lambda, s, \log w, \log t)$.

Definition 3.2.13 (Knowledge Soundness)**.** *Let* $\kappa : \mathbb{N}^4 \to [0, 1]$*. A system* Π *is said to be* $(\kappa, \mathcal{X}^*)$*-knowledge-sound if for any PPT adversary* $\mathcal{A}$ *there exists a PPT extractor* $\mathcal{E}_\mathcal{A}$

such that the following probability is at most $\kappa(\lambda)$:

$$\Pr\left[\begin{array}{l|l} \Big(\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, \pi) = 1\Big) \wedge & \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t) \\ \Big((f, y, \mathbf{z}) \notin \mathcal{F}_{s,w,t} \times \mathcal{Y}_{s,t} \times \mathcal{X}^s\Big) & ((f, y, \mathbf{z}), \pi) \leftarrow \mathcal{A}(\mathsf{pp}; r_{\mathcal{A}}) \\ \vee\ (\mathbf{x} \notin (\mathcal{X}^*)^w) & \mathbf{x} \leftarrow \mathcal{E}_{\mathcal{A}}(\mathsf{pp}, r_{\mathcal{A}}) \\ \vee\ f(\mathbf{z}, \mathbf{x}) \neq y(\mathbf{z})\Big) & \mathsf{pp}_{f,y} \leftarrow \mathsf{PreVerify}(\mathsf{pp}, (f, y)) \end{array}\right]$$

We say that the SNARK is $\mathcal{X}^*$-knowledge-sound if it is $(\kappa, \mathcal{X}^*)$-knowledge-sound and $\kappa(\lambda, s, w, t)$ is a negligible in λ for any $s, w, t \in \mathsf{poly}(\lambda)$.

Definition 3.2.14 (Preprocessing SNARKs)**.** *A preprocessing non-interactive argument system Π is said to be a preprocessing SNARK if it is complete, succinct, and $\mathcal{X}^*$-knowledge-sound for some $\mathcal{X}^* \supseteq \mathcal{X}$.*

Sometimes SNARKs are required to be zero-knowledge (zk-SNARKs), in which case we also require the existence of a simulator that is able to generate valid proofs without knowing the witness. Contrary to standard zero-knowledge proofs, SNARKs are already non-trivial to construct without zero-knowledge, so we treat this aspect as tangential to our main result. We refer the reader to Definition B.1.7 for a formal definition of this property.

3.3 The k-M-ISIS Assumption

We first introduce a family of assumptions over modules – k-M-ISIS – which we then specialise to rings to obtain k-R-ISIS mentioned above.

We note that the most immediate candidate notion for k-ISIS, i.e. generalising k-SIS, is to simply hand out short preimages of random images and then ask the adversary to solve ISIS. This notion is trivially equivalent to ISIS since short preimages of random images can be efficiently sampled by sampling short $\mathbf{u} \in \mathbb{Z}^\ell$ and computing $\mathbf{t} \coloneqq \mathbf{A} \cdot \mathbf{u} \bmod q$. The same reasoning can be lifted to $\mathcal{R}$. On the other hand, k-SIS is trivially insecure when $k \geq \ell$ in the intuitive sense since then $\{\, \mathbf{u}_i \,\}$ constitutes a trapdoor for $\mathbf{A}$ when the $\mathbf{u}_i$ are linearly independent [GPV08]. Formally, the problem as stated is impossible to solve since all vectors will be in $\mathbb{Q}\text{-}\operatorname{span}(\{\, \mathbf{u}_i \,\}_{0 \leq i < k})$, i.e. there are no valid solutions.

Our variants are neither trivially equivalent to M-ISIS nor immediately broken when $k > \ell$ by imposing on the images an algebraic structure which is independent of the challenge matrix $\mathbf{A}$. Before stating our family of assumptions, we define a notion of admissibility to formally rule out trivial wins.

Definition 3.3.1 (k-M-ISIS-Admissible)**.** *Let $g(\mathbf{X}) \in \mathcal{R}(\mathbf{X})$ be a Laurent monomial, i.e. $g(\mathbf{X}) = \mathbf{X}^{\mathbf{e}} \coloneqq \prod_{i \in \mathbb{Z}_w} X_i^{e_i}$ for some exponent vector $\mathbf{e} = (e_i : i \in \mathbb{Z}_w) \in \mathbb{Z}^w$. Let $\mathcal{G} \subset \mathcal{R}(\mathbf{X})$ be a set of Laurent monomials with $k \coloneqq |\mathcal{G}|$ and let $\boldsymbol{\mathcal{G}}$ be a vector of those*

monomials. Let $g^* \in \mathcal{R}(\mathbf{X})$ be a target Laurent monomial. We call a family $\mathcal{G}$ k-M-ISIS-admissible if 1. all $g \in \mathcal{G}$ have constant degree, i.e. $\|\mathbf{e}\|_1 \in O(1)$; 2. all $g \in \mathcal{G}$ are distinct, i.e. $\mathcal{G}$ is not a multiset; and 3. $0 \notin \mathcal{G}$. We call a family $(\mathcal{G}, g^*)$ k-M-ISIS-admissible if $\mathcal{G}$ is k-M-ISIS-admissible, g^* has constant degree, and $g^* \notin \mathcal{G}$.

Remark 4. *Condition (i) rules out monomials that depend on the ring $\mathcal{R}$, such as $X^{\phi(m)}$. Condition (ii) rules out that trivial linear combinations of known preimages produce a preimage for the target. Condition (iii) rules out trivially producing multiple preimages of the same image. On the other hand, we do not target full generality here but restrict ourselves to a slight generalisation of what we require in this work. It is plausible that we can replace Laurent monomials by Laurent "terms", i.e. with coefficients $\neq 1$ in $\mathcal{R}_q$, or rational functions.*

Definition 3.3.2 (k-M-ISIS Assumptions)**.** *Let $\ell, \eta \in \mathbb{N}$. Let q be a rational prime, $\mathcal{R}$ the m-th cyclotomic ring, and $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. Let $\mathcal{T} \subset \mathcal{R}_q^\eta$ be such that, for any $\mathbf{t} = (t_i)_{i \in \mathbb{Z}_\eta} \in \mathcal{T}$, $\langle \{ t_i \} \rangle = \mathcal{R}_q$. Let $\mathcal{G} \subset \mathcal{R}(\mathbf{X})$ be a set of w-variate Laurent monomial. Let $g^* \in \mathcal{R}(\mathbf{X})$ be a target Laurent monomial. Let $(\mathcal{G}, g^*)$ be k-M-ISIS-admissible. Let $\bar{\mathcal{G}} := \mathcal{G} \cup \{ g^* \}$. Let $\beta \geq 1$ and $\beta^* \geq 1$ be reals. For $\eta, \ell \in \mathbb{N}$, $g \in \bar{\mathcal{G}}$, $\ell \geq \mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$, $\mathbf{A} \in \mathcal{R}_q^{\eta \times \ell}$, $\mathbf{t} \in \mathcal{T}$, and $\mathbf{v} \in (\mathcal{R}_q^\times)^w$, let $\mathcal{D}_{g,\mathbf{A},\mathbf{t},\mathbf{v}}$ be a distribution over*

$$\{ \mathbf{u}_g \in \mathcal{R}^\ell : \mathbf{A} \cdot \mathbf{u}_g \equiv g(\mathbf{v}) \cdot \mathbf{t} \bmod q, \|\mathbf{u}_g\| \leq \beta \}.$$

Let $\mathcal{D} := \{ \mathcal{D}_{g,\mathbf{A},\mathbf{t},\mathbf{v}} : \eta, \ell \in \mathbb{N}, g \in \bar{\mathcal{G}}, \mathbf{A} \in \mathcal{R}_q^{\eta \times \ell}, \mathbf{v} \in (\mathcal{R}_q^\times)^w \}$ be the family of these distributions. Write $\mathsf{pp} := (\mathcal{R}_q, \eta, \ell, w, \mathcal{G}, g^*, \mathcal{D}, \mathcal{T}, \beta, \beta^*)$. The k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ assumption states that for any PPT adversary $\mathcal{A}$ we have $\mathsf{Adv}^{\mathsf{k\text{-}m\text{-}isis}}_{\mathsf{pp},\mathcal{A}} \leq \mathsf{negl}(\lambda)$, where

$$\mathsf{Adv}^{\mathsf{k\text{-}m\text{-}isis}}_{\mathsf{pp},\mathcal{A}} := \Pr\left[\begin{array}{l|l} \mathbf{A} \cdot \mathbf{u}_{g^*} \equiv s^* \cdot g^*(\mathbf{v}) \cdot \mathbf{t} \bmod q & \mathbf{A} \leftarrow \mathcal{R}_q^{\eta \times \ell} \bmod q \\ \wedge\ 0 < \|s^*\| \leq \beta^* & \mathbf{t} \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w \\ \wedge\ \|\mathbf{u}_{g^*}\| \leq \beta^* & \mathbf{u}_g \leftarrow \mathcal{D}_{g,\mathbf{A},\mathbf{t},\mathbf{v}},\ \forall\ g \in \mathcal{G} \\ \wedge\ (g^*, \mathbf{u}_{g^*}) \neq (0, \mathbf{0}) & (s^*, \mathbf{u}_{g^*}) \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{t}, \{ \mathbf{u}_\mathcal{G} \}, \mathbf{v}) \end{array}\right].$$

Remark 5. *Since for any $\mathbf{t}' \in \mathcal{T}$ there exist matrices $\mathbf{X}, \mathbf{Y}$ s.t. $\mathbf{X} \cdot \mathbf{Y} \equiv \mathbf{I}$, $\mathbf{X} \cdot \mathbf{t}' \equiv (1, 0, \ldots, 0)^\mathsf{T} \bmod q$ and $\mathbf{Y} \cdot (1, 0, \ldots, 0)^\mathsf{T} \equiv \mathbf{t}' \bmod q$, we can assume that $\mathcal{T} = \{ (1, 0, \ldots, 0)^\mathsf{T} \}$ without loss of generality.*

Definition 3.3.3 (k-R-ISIS)**.** *When $\eta = 1$ we may write*

$$k\text{-}R\text{-}\mathsf{ISIS}_{\mathcal{R}_q,\ell,w,\mathcal{G},g^*,\mathcal{D},\mathcal{T},\beta,\beta^*} := k\text{-}M\text{-}\mathsf{ISIS}_{\mathcal{R}_q,1,\ell,w,\mathcal{G},g^*,\mathcal{D},\mathcal{T},\beta,\beta^*}.$$

Remark 6. *Analogous to the ℓ-Diffie-Hellman exponent assumption, an example of $(w, \mathcal{G}, g^*)$ is $w = 1$, $\mathcal{G} = \{ 1, X, \ldots, X^\ell, X^{\ell+2}, \ldots, X^{2\ell} \}$, and $g^*(X) = X^{\ell+1}$ for some $\ell \in \mathbb{N}$.*

As written above we have a separate assumption for each family of $(\mathcal{G}, g^*)$ which are application dependent. As we will show below, there are $(\mathcal{G}, g^*)$ that are as hard as

M-ISIS and our discussion of admissibility indicates that some $(\mathcal{G}, g^*)$ are trivially insecure. However, to encourage analysis and to avoid "bodacious assumptions" [KM10] we make the following, strong, meta assumption.

Definition 3.3.4 (k-M-ISIS Meta Assumption)**.** *For any k-M-ISIS-admissible $(\mathcal{G}, g^*)$, k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ with $\mathsf{pp} := (\mathcal{R}_q, \eta, \ell, w, \mathcal{G}, g^*, \mathcal{D}, \mathcal{T}, \beta, \beta^*)$ is hard if M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta,\ell,\beta^*}$ is hard.*

3.3.1 Knowledge Variants

We next propose a "knowledge" version of the k-M-ISIS assumption. It captures the intuition that if the images are restricted to scalar multiples of $\mathbf{t}$ then the only way to produce preimages of them under $\mathbf{A}$ is to perform a linear combination of the given preimages under $\mathbf{A}$ with small coefficients.

Definition 3.3.5 (Knowledge k-M-ISIS Assumption)**.** *Adopt the notation from Definition 3.3.2, but let $\mathsf{pp} := (\mathcal{R}_q, \eta, \ell, w, \mathcal{G}, \mathcal{D}, \mathcal{T}, \alpha^*, \beta, \beta^*)$ where $\alpha^* \geq 1$ is real and $\eta > 1$. The knowledge k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ assumption states that for any PPT adversary $\mathcal{A}$ there exists a PPT extractor $\mathcal{E}_{\mathcal{A}}$ such that $\mathsf{Adv}^{\mathsf{know\text{-}k\text{-}m\text{-}isis}}_{\mathsf{pp},\mathcal{A}} \leq \mathsf{negl}(\lambda)$, where*

$$\mathsf{Adv}^{\mathsf{know\text{-}k\text{-}m\text{-}isis}}_{\mathsf{pp},\mathcal{A}} := \Pr\left[\begin{array}{l|l} \mathbf{A}\cdot\mathbf{u} \equiv c\cdot\mathbf{t} \bmod q & \mathbf{A}\leftarrow\mathcal{R}_q^{\eta\times\ell} \\ \wedge\ \|\mathbf{u}\| \leq \beta^* & \mathbf{t}\leftarrow\mathcal{T};\ \mathbf{v}\leftarrow(\mathcal{R}_q^{\times})^w \\ \wedge\ \neg\left(\begin{array}{l} c \equiv \sum_{g\in\mathcal{G}} x_g\cdot g(\mathbf{v}) \bmod q \\ \wedge\ \left\|(x_g)_{g\in\mathcal{G}}\right\| \leq \alpha^* \end{array}\right) & \mathbf{u}_g\leftarrow\mathcal{D}_{g,\mathbf{A},\mathbf{t},\mathbf{v}},\ \forall\ g\in\mathcal{G} \\ & \left((c,\mathbf{u}),(x_g)_{g\in\mathcal{G}}\right) \\ & \quad\leftarrow(\mathcal{A}\|\mathcal{E}_{\mathcal{A}})\left(\mathbf{A},\mathbf{t},\{\,\mathbf{u}_{\mathcal{G}}\,\},\mathbf{v}\right) \end{array}\right]$$

where the notation $(\mathcal{A}\|\mathcal{E}_{\mathcal{A}})$ means that $\mathcal{A}$ and $\mathcal{E}_{\mathcal{A}}$ are run on the same input including the randomness, and $(c,\mathbf{u})$ and $(x_g)_{g\in\mathcal{G}}$ are the outputs of $\mathcal{A}$ and $\mathcal{E}_{\mathcal{A}}$ respectively.

The knowledge k-M-ISIS assumption, as stated, only makes sense for $\eta \geq 2$, i.e. not for k-R-ISIS. To see this, consider an adversary $\mathcal{A}$ which does the following: First, it samples random short $\mathbf{u}$ and checks whether $\mathbf{A}\cdot\mathbf{u} \bmod q$ is in the submodule of $\mathcal{R}_q^{\eta}$ generated by $\mathbf{t}$. If not, $\mathcal{A}$ aborts. If so, it finds c such that $\mathbf{A}\cdot\mathbf{u} = c\cdot\mathbf{t} \bmod q$ and outputs $(c,\mathbf{u})$. When $\eta = 1$ and assuming without loss of generality that $\mathcal{T} = \{\,(1,0,\ldots,0)^{\mathsf{T}}\,\}$, we observe that $t = 1$ generates $\mathcal{R}_q$, which means $\mathcal{A}$ never aborts. Clearly, when $\mathcal{A}$ does not abort, it has no "knowledge" of how c can be expressed as a linear combination of $\{\,g(\mathbf{v})\,\}_{g\in\mathcal{G}}$. Note that when $\eta \geq 2$ the adversary $\mathcal{A}$ aborts with overwhelming probability since $\mathbf{A}\cdot\mathbf{u} \bmod q$ is close to uniform over $\mathcal{R}_q^{\eta}$ but the submodule generated by $\mathbf{t}$ is only a negligible faction of $\mathcal{R}_q^{\eta}$. However, in order to be able to pun about "crises of knowledge", we also define a ring version of the knowledge assumption. In the ring setting, we consider proper ideals rather than submodules.

Definition 3.3.6 (Knowledge k-R-ISIS Assumption)**.** *Let the parameters pp be as in Definition 3.3.2 except that $\eta = 1$ and $\mathcal{T}$ contains elements $t \in \mathcal{R}_q$ s.t. $1/|\langle t\rangle| = \mathsf{negl}$*

and $|\langle t\rangle|/|\mathcal{R}_q| = \mathsf{negl}$. *Furthermore, let* $\mathcal{S}_t \coloneqq \{s \in \mathcal{R}_q \mid s \cdot t \equiv 0 \bmod q\}$ *and let* $\mathcal{T}$ *be such that finding* $s' \in \mathcal{S}_t$ *with* $\|s'\| \leq \alpha^*$ *is hard for* $t \leftarrow \mathcal{T}$.[6] *The knowledge* k-R-$\mathsf{ISIS}_{\mathsf{pp}}$ *assumption states that for any PPT adversary* $\mathcal{A}$ *there exists a PPT extractor* $\mathcal{E}_{\mathcal{A}}$ *such that* $\mathsf{Adv}^{\mathsf{k\text{-}r\text{-}isis}}_{\mathsf{pp},\mathcal{A}} \leq \mathsf{negl}(\lambda)$, *where*

$$\mathsf{Adv}^{\mathsf{k\text{-}r\text{-}isis}}_{\mathsf{pp},\mathcal{A}} \coloneqq \Pr\left[\begin{array}{l} \langle \mathbf{a}, \mathbf{u}\rangle \equiv c \cdot t \bmod q \\ \wedge\ \|\mathbf{u}\| \leq \beta^* \\ \wedge\ \neg\left(\begin{array}{l} c \equiv \sum_{g\in\mathcal{G}} x_g \cdot g(\mathbf{v}) \bmod q \\ \wedge\ \left\|(x_g)_{g\in\mathcal{G}}\right\| \leq \alpha^* \end{array}\right) \end{array} \middle| \begin{array}{l} \mathbf{a} \leftarrow \mathcal{R}_q^{\ell} \\ t \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^{\times})^{w} \\ \mathbf{u}_g \leftarrow \mathcal{D}_{g,\mathbf{a},t,\mathbf{v}},\ \forall\ g \in \mathcal{G} \\ \left((c,\mathbf{u}),(x_g)_{g\in\mathcal{G}}\right) \\ \quad \leftarrow (\mathcal{A}\|\mathcal{E}_{\mathcal{A}})\left(\mathbf{a}, t, \{\,\mathbf{u}_{\mathcal{G}}\,\}, \mathbf{v}\right) \end{array}\right].$$

Definition 3.3.7 (Knowledge k-M-ISIS Meta Assumption)**.** *Let* $(\mathcal{G}, g^*)$ *be* k-M-ISIS*-admissible,* α^*, β^* *be reals with* $\alpha^* \geq \beta^* \geq 1$, *and* $\eta > 1$. *The knowledge* k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta,\ell,w,\mathcal{G},\mathcal{D},\mathcal{T},\alpha^*,\beta,\beta^*}$ *assumption holds if the* k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta,\ell,w,\mathcal{G},g^*,\mathcal{D},\mathcal{T},\beta,\beta^*}$ *assumption holds.*

Remark 7. *We note that our meta assumption does not cover knowledge* k-R-ISIS *since it is not a true special case of* k-M-ISIS *as discussed above. See also discussion in Section 3.4.2.*

3.4 Analysing the k-M-ISIS Assumption

We give reductions studying the properties of our new assumptions. We first show that there exist hard instances of the k-R-ISIS problem. In particular, in Lemmas 3.4.1 and 3.4.2 we show that k-R-ISIS (with $g^* \equiv 1$) is as hard as R-SIS when $w \geq k$ and when the system generated by $\mathcal{G}$ is efficiently invertible. In both lemmas, we use that $g(\mathbf{v}) \sim U((\mathcal{R}_q^{\times})^{w}) \approx U(\mathcal{R}_q^{w})$ and these reductions do not apply to k-M-ISIS, i.e. $\eta > 1$. In Theorem 3.4.1 we show that k-M-ISIS is at least as hard as k-R-ISIS. This, on the one hand, formalises the intuition that increasing the module rank does not make the problem easier but, on the other hand, also shows that the additional structure (restricting to multiples of $\mathbf{t} \coloneqq (1, 0, \ldots, 0)^{\mathsf{T}}$) preserves hardness. Using the same techniques, in Theorem 3.4.2 we also show that k-M-ISIS is a true generalisation of k-R-SIS. We stress, however, that none of the above reductions cover the case we use for our example application in Section 3.5.

We next study the relations between k-M-ISIS (but not just k-R-ISIS) problems for different choices of $(\mathcal{G}, g^*)$. In Lemma 3.4.4 we show that $(\mathcal{G}, g^*)$ is as hard as $(\mathcal{G}, 0)$ for any $\mathcal{G}$, formalising the intuition that the non-homogeneous variant is no easier than the homogeneous variant. Then, in Lemma 3.4.5 we show that scaling $(\mathcal{G}, g^*)$ multiplicatively

[6] Concretely, let $\mathcal{T}$ be the set of all $\mathcal{R}_q$ elements t where half of the components of t in the Chinese remainder theorem (CRT) representation are zero and the other half are non-zero. Note that this is well-defined only when $\langle q\rangle$ is not a prime ideal in $\mathcal{R}$. See Section 3.4.2 for more discussion on the choices for $(\mathcal{R}_q, \mathcal{T})$.

by any non-zero Laurent monomial does not change the hardness, e.g. we may choose to normalise instances to $g^* \equiv 1$.

Finally, in Section 3.4.1, we investigate attacks on the k-M-ISIS problem. These attacks do not outperform standard attacks on SIS and we will use them to set parameters in Section 3.5.1.

Some k-R-ISIS $\geq$ R-SIS.

First, we show that giving out up to w constraints and when $g^* \equiv 1$ then k-R-ISIS is no easier than R-SIS. Under this condition, we can simply sample random preimages and solve for the right $\mathbf{v}$ to satisfy the $\mathcal{G}$ constraints.

Lemma 3.4.1. *Let the parameters* pp *be as in Definition 3.3.2. Furthermore, let* $g^* \equiv 1$, $\mathcal{G} = \{g_i(\mathbf{X})\}_{i\in\mathbb{Z}_k} \subset \mathcal{R}(\mathbf{X})$ *be of size* $k \leq w$*, the number of variables, and* $\mathcal{D}$ *be such that the distribution*

$$\left\{(\mathbf{a}, t, \{\mathbf{u}_i\}, \mathbf{v}) \middle| \mathbf{a}\leftarrow\mathcal{R}_q^\ell;\ t\leftarrow\mathcal{T};\ \mathbf{v}\leftarrow(\mathcal{R}_q^\times)^w;\ \mathbf{u}_i\leftarrow\mathcal{D}_{g_i,\mathbf{a},t,\mathbf{v}},\ \forall i\in\mathbb{Z}_k\right\}$$

is statistically close to the distribution

$$\left\{(\mathbf{a}, t, \{\mathbf{u}_i\}, \mathbf{v}) \middle| \begin{array}{l}\mathbf{a}\leftarrow\mathcal{R}_q^\ell;\ t\leftarrow\mathcal{T};\ \mathbf{v}\leftarrow(\mathcal{R}_q^\times)^w \\ \mathbf{u}_i\leftarrow\mathsf{SampD}(1^1, 1^\ell, \mathcal{R}, \beta) : \langle\mathbf{a}, \mathbf{u}_i\rangle \equiv g_i(\mathbf{v})\cdot t \bmod q,\ \forall i\in\mathbb{Z}_k\end{array}\right\}.$$

Write $g_i(\mathbf{X}) = \mathbf{X}^{\mathbf{e}_i}$, $\mathbf{E} = (\mathbf{e}_i)_{i\in\mathbb{Z}_k} \in \mathbb{Z}^{k\times k}$, *and* $(g_i(\mathbf{v}))_{i\in\mathbb{Z}_k} = \mathbf{v}^{\mathbf{E}}$. *If* $\mathcal{R}_q$ *is a field, let* $\gcd(\det(\mathbf{E}), q^n - 1) = 1$. *Otherwise let* $\det(\mathbf{E}) = \pm 1$. *Then for any PPT adversary* $\mathcal{A}$ *against* k-R-$\mathsf{ISIS}_{\mathsf{pp}}$ *there exists a PPT adversary* $\mathcal{A}'$ *against* R-SIS *with*

$$\mathsf{Adv}^{\text{r-sis}}_{\mathcal{R}_q,\ell+1,\beta^*,\mathcal{A}'} \quad\geq\quad \frac{1}{\mathsf{poly}(\lambda)}\cdot\mathsf{Adv}^{\text{k-r-isis}}_{\mathsf{pp},\mathcal{A}}.$$

Proof. Wlog we consider $k = w$ by simply only submitting a subset of our preimages to the adversary. Also wlog we assume $\mathcal{T} = \{1\}$ as discussed in Remark 5. We construct an R-SIS solver as follows: On input of an R-SIS instance $\mathbf{a}'$, write $\mathbf{a}' = (\bar{\mathbf{a}}, a')$ and set $\mathbf{a} = \frac{1}{a'}\cdot\bar{\mathbf{a}}$. If no a' is invertible in $\mathcal{R}_q$ the reduction aborts. By our choice of q, with non-negligible probability over the randomness of $\mathbf{a}'$ the reduction does not abort, and in which case $\mathbf{a}$ is uniformly distributed over $\mathcal{R}_q^\ell$. For $i \in \mathbb{Z}_k$ sample $\mathbf{u}_i\leftarrow\mathsf{SampD}(1^1, 1^\ell, \mathcal{R}, \beta)$ and compute $t_i = \langle\mathbf{a}, \mathbf{u}_i\rangle$. Since $\ell \geq \mathsf{lhl}(\mathcal{R}, 1, q, \beta)$, $\mathbf{t} \in \mathcal{R}_q^k$ is distributed within negligible statistical distance to uniform. By our choice of q, we have $\mathbf{t} \in (\mathcal{R}_q^\times)^k$ with non-negligible probability. Compute $\mathbf{v} = (\mathbf{t})^{\mathbf{E}^{-1}}$. We can write $\mathbf{E}^{-1}$ because $\mathbf{E}^{-1} = \mathbf{F}/r$ where $\mathbf{F}$ is over $\mathbb{Z}$ and $r := |\det(\mathbf{E})| \in \mathbb{Z}$. If $\det(\mathbf{E}) = \pm 1$ compute $\mathbf{v}$ directly. Otherwise, note that every element in a finite field of order q^n has an r-th root if $\gcd(r, q^n - 1) = 1$ and computing r-th roots can be accomplished by computing $r^{-1} \bmod (q^n - 1)$. Note that this implies r-th roots are unique under these conditions and the map is a bijection. Thus, the map defined by $g_i(\mathbf{X})$ is a bijection, implying our sampling procedure produces well distributed inputs.

Run the k-R-SIS solver on $(\mathbf{a}, \{\mathbf{u}_i\}_{i\in\mathbb{Z}_k}, \mathbf{v})$ to obtain $(\mathbf{u}^*, s^*)$ satisfying $\langle \mathbf{a}, \mathbf{u}^* \rangle \equiv s^* \bmod q$. Output $\mathbf{u}' = (\mathbf{u}^*, -s^*)$. We observe that

$$\begin{aligned} \langle \mathbf{a}, \mathbf{u}^* \rangle &\equiv s^* \bmod q \\ \langle (\mathbf{a}, 1), (\mathbf{u}^*, -s^*) \rangle &\equiv 0 \bmod q \\ \langle a' \cdot (\mathbf{a}, 1), (\mathbf{u}^*, -s^*) \rangle &\equiv 0 \bmod q \\ \langle \mathbf{a}', \mathbf{u}' \rangle &\equiv 0 \bmod q \end{aligned}$$

Our R-SIS solver runs in time proportional to our k-R-SIS solver. Finally, observe that $\|\mathbf{u}'\| \leq \beta^*$ if the k-M-SIS adversary succeeded. □

Next, we show that for some additional forms of $\mathcal{G}$, too, k-R-SIS is equivalent to R-SIS. Here we use the freedom to sample v_i to fix up images.

Lemma 3.4.2. *Let the parameters* pp *be as in Definition 3.3.2. Furthermore, let* $w = w' + k$ *for some* $w' \in \mathbb{N}$, $\mathcal{G}$ *be of the form*

$$\mathcal{G} = \{ g_i(\mathbf{X}) \}_{i\in\mathbb{Z}_k} = \{ X_{w'+i} \cdot \prod_{j\in\mathbb{Z}_{w'}} X_j^{e_j} \}_{i\in\mathbb{Z}_k},$$

and $\mathcal{D}$ *be such that the distribution*

$$\left\{ (\mathbf{a}, t, \{\mathbf{u}_i\}, \mathbf{v}) \middle| \mathbf{a} \leftarrow \mathcal{R}_q^\ell;\ t \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w;\ \mathbf{u}_i \leftarrow \mathcal{D}_{g_i,\mathbf{a},t,\mathbf{v}},\ \forall i \in \mathbb{Z}_k \right\}$$

is statistically close to the distribution

$$\left\{ (\mathbf{a}, t, \{\mathbf{u}_i\}, \mathbf{v}) \middle| \begin{array}{l} \mathbf{a} \leftarrow \mathcal{R}_q^\ell;\ t \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w \\ \mathbf{u}_i \leftarrow \mathsf{SampD}(1^1, 1^\ell, \mathcal{R}, \beta) : \langle \mathbf{a}, \mathbf{u}_i \rangle \equiv g_i(\mathbf{v}) \cdot t \bmod q,\ \forall i \in \mathbb{Z}_k \end{array} \right\}.$$

For any PPT adversary $\mathcal{A}$ *against* k-R-$\mathsf{SIS}_{\mathsf{pp}}$ *there exists a PPT adversary* $\mathcal{A}'$ *against* R-SIS *with*

$$\mathsf{Adv}^{\text{r-sis}}_{\mathcal{R}_q,\ell,\beta^*,\mathcal{A}'} \geq \frac{1}{\mathsf{poly}(\lambda)} \cdot \mathsf{Adv}^{\text{k-r-isis}}_{\mathsf{pp},\mathcal{A}}.$$

Proof. Let $\mathbf{a}$ be a R-$\mathsf{SIS}_{\mathcal{R}_q,\ell,\beta^*}$ instance. By assumption, $\mathbf{a}$ is uniformly distributed over $\mathcal{R}_q^\ell$. Sample $\mathbf{v} \leftarrow (\mathcal{R}_q^\times)^{w'}$ and $\mathbf{u}_i \leftarrow \mathsf{SampD}(1^1, 1^\ell, \mathcal{R}, \beta)$ for all $i \in \mathbb{Z}_k$. Compute $y_i \equiv \langle \mathbf{a}, \mathbf{u}_i \rangle \bmod q$ and $v_{w'+i} \equiv y_i \cdot \prod_{j\in\mathbb{Z}_{w'}} v_j^{-e_j} \bmod q$ for all $i \in \mathbb{Z}_k$.

If y_i is not invertible for any $i \in \mathbb{Z}_k$ the reduction aborts. If the reduction does not abort, which happens with non-negligible probability, since $\ell \geq \mathsf{lhl}(\mathcal{R}, 1, q, \beta)$, for each $i \in \mathbb{Z}_k$, y_i is uniformly distributed over $\mathcal{R}_q^\times$, and so is $v_{w'+i}$. We therefore conclude that $\mathbf{v}$ is uniformly distributed over $(\mathcal{R}_q^\times)^w$. Run the k-R-SIS adversary on the input $\left(\mathbf{a}, \{\mathbf{u}_i\}_{i\in\mathbb{Z}_k}, \mathbf{v}\right)$ to obtain $(s^*, \mathbf{u}_{g^*})$. By construction $\mathbf{u}_{g^*}$ satisfies $\langle \mathbf{a}, \mathbf{u}_{g^*} \rangle \equiv 0 \bmod q$ if the k-R-SIS adversary succeeded. □

k-M-ISIS $\geq$ k-R-(I)SIS.

We show that k-M-ISIS is no easier than k-R-ISIS. The analogous reduction for M-ISIS and R-ISIS is trivial. Here we face the complication that we have to map the known preimages to k-M-ISIS while preserving a mapping back to make use of the returned k-M-ISIS solution in k-R-ISIS. We do this by constructing a lower-triangular matrix that satisfies our constraints and hide its structure by multiplying with a short upper triangular matrix (with a short inverse). We then use Rényi divergence arguments to break thus introduced dependencies. Our reduction has several limitations: 1. It requires $\ell \geq \mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$ rather than $\ell > \mathsf{lhl}(\mathcal{R}, 1, q, \beta)$ for the input k-R-ISIS instance and 2. it produces an output distribution $\mathcal{D}$ for k-M-ISIS that is non-spherical. For ease of exposition and because we do not require the more general case in this work, we give our reduction for $\eta = 2$.

Theorem 3.4.1. *Let the parameters* pp_M *and* pp_R *for* k-M-ISIS *and* k-R-ISIS *respectively be as in Definition 3.3.2, such that they share the same ring* $\mathcal{R}_q$*, number of variables* w*, and monomials* $(\mathcal{G}, g^*)$*. Differing parameters are distinguished by subscripts, e.g.* ℓ_M *and* ℓ_R*. Furthermore, let* $\eta_M = 2$*,* $\beta^*_\Delta \in \mathbb{R}$*,* $\sigma, \sigma_\Delta > \eta_\epsilon(\mathcal{R}) \in \mathbb{R}$*,* $\beta_x \geq \sigma_x$ *be s.t.* $u \sim \mathcal{D}_{\mathcal{R}^\ell, \sigma_x}$ *satisfy* $\|u\|_\infty \leq \beta_x$ *for* $x \in \{ R, M, \Delta \}$*,* $\ell_\Delta := \ell_M - \ell_R \geq \mathsf{lhl}(\mathcal{R}, 1, q, \beta_\Delta)$*,* $\sigma_R > \gamma_\mathcal{R} \cdot (\ell_\Delta \cdot n)^{3/2} \cdot \gamma_\mathcal{R} \cdot \sigma_\Delta \cdot \sigma$*,* $\beta^*_R \geq 2\,\ell_\Delta \cdot \gamma_\mathcal{R} \cdot \sqrt{n} \cdot \sigma \cdot \beta^*_\Delta$*,* $\ell_R \geq \mathsf{lhl}(\mathcal{R}, 1, q, \sigma)$ *and* $\geq \mathsf{lhl}(\mathcal{R}, 2, q, \beta_R)$*. Let* $\mathcal{D}_R$ *be such that the distribution*

$$\left\{ (\mathbf{a}, t, \{ \mathbf{u}_i \}, \mathbf{v}) \middle| \mathbf{a} \leftarrow \mathcal{R}_q^{\ell_R};\ t \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w;\ \mathbf{u}_i \leftarrow \mathcal{D}_{g_i, \mathbf{a}, \mathbf{v}},\ \forall i \in \mathbb{Z}_k \right\}$$

is statistically close to the distribution

$$\left\{ (\mathbf{a}, t, \{ \mathbf{u}_i \}, \mathbf{v}) \middle| \begin{array}{l} \mathbf{a} \leftarrow \mathcal{R}_q^{\ell_R};\ t \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w \\ \mathbf{u}_i \leftarrow \mathcal{D}_{\mathcal{R}^{\ell_R}, \sigma_R} : \langle \mathbf{a}, \mathbf{u}_i \rangle \equiv g_i(\mathbf{v}) \cdot t \bmod q,\ \forall i \in \mathbb{Z}_k \end{array} \right\}.$$

Let $\mathcal{D}_M$ *be such that the distribution*

$$\left\{ (\mathbf{A}, \mathbf{t}, \{ \mathbf{u}_i \}, \mathbf{v}) \middle| \mathbf{A} \leftarrow \mathcal{R}_q^{\eta_M \times \ell_M};\ \mathbf{t} \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w;\ \mathbf{u}_i \leftarrow \mathcal{D}_{g_i, \mathbf{A}, \mathbf{t}, \mathbf{v}},\ \forall i \in \mathbb{Z}_k \right\}$$

is statistically close to the distribution

$$\left\{ (\mathbf{A}, \mathbf{t}, \{ \mathbf{u}_i \}, \mathbf{v}) \middle| \begin{array}{l} \mathbf{A} \leftarrow \mathcal{R}_q^{\eta_M \times \ell_M};\ \mathbf{t} \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w \\ \mathbf{u}_i \leftarrow \mathcal{D}_{\mathcal{R}^{\ell_R}, \sigma_R} \times \mathcal{D}_{\mathcal{R}^{\ell_\Delta}, \sigma_\Delta} : \mathbf{A} \cdot \mathbf{u}_i \equiv g_i(\mathbf{v}) \cdot \mathbf{t} \bmod q,\ \forall i \in \mathbb{Z}_k \end{array} \right\}.$$

Let SampD *and* SampPre *output samples following a Discrete Gaussian distribution of appropriate width* σ_x *given* β_x*. For any PPT adversary* $\mathcal{A}$ *against* k-M-$\mathsf{ISIS}_{\mathsf{pp}_M}$ *there exists a PPT adversary* $\mathcal{A}'$ *against* k-R-$\mathsf{ISIS}_{\mathsf{pp}_R}$ *with*

$$\mathsf{Adv}^{\text{k-r-isis}}_{\mathsf{pp}_R, \mathcal{A}'} \quad \geq \quad \frac{1}{\mathsf{poly}(\lambda)} \cdot \mathsf{Adv}^{\text{k-m-isis}}_{\mathsf{pp}_M, \mathcal{A}}.$$

Using the same proof strategy, we show that some k-M-ISIS adversaries can break k-M-SIS. To ease readability, the formal statement below is for k-R-SIS, i.e. k-M-SIS with $\eta = 1$. The only non-trivial step is to argue that the output solution satisfies the additional constraint imposed by k-M-SIS. Here we use an unrelated R-SIS instances to argue that the adversary either broke R-SIS or the solution satisfies the required constraint that it is not in $\mathcal{K}\text{-}\operatorname{span}(\{\mathbf{u}_i\}_{0\leq i<k})$.

Theorem 3.4.2. *Let the parameters pp_M for k-M-ISIS be as in Definition 3.3.2. Furthermore, let $\eta_M = 2$, $g_M^* = 0$, $\beta_\Delta^* \in \mathbb{R}$, $\sigma, \sigma_\Delta > \eta_\epsilon(\mathcal{R}) > 1 \in \mathbb{R}$, $\beta_x \geq \sigma_x$ be s.t. $u \sim \mathcal{D}_{\mathcal{R},\sigma_x}$ satisfy $\|u\|_\infty \leq \beta_x$ for $x \in \{R, M, \Delta\}$, $\ell_\Delta \coloneqq \ell_M - \ell_R \geq \mathsf{lhl}(\mathcal{R}, 1, q, \beta_\Delta)$, $\sigma_R > (4\,\gamma_\mathcal{R} \cdot \sigma_\Delta \cdot \sigma \cdot \ell_\Delta \cdot n)$, $\beta_R^* \geq 2\,\ell_\Delta \cdot \gamma_\mathcal{R} \cdot \sqrt{2\,\pi n} \cdot \sigma \cdot \beta_\Delta^*$, $\ell_R \geq \mathsf{lhl}(\mathcal{R}, 1, q, \sigma)$ and $\geq \mathsf{lhl}(\mathcal{R}, 2, q, \beta_R)$. Let $\mathcal{D}_M$ be such that the distribution*

$$\left\{(\mathbf{A}, \mathbf{t}, \{\mathbf{u}_i\}, \mathbf{v}) \middle| \mathbf{A} \leftarrow \mathcal{R}_q^{\eta_M \times \ell_M};\ \mathbf{t} \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w;\ \mathbf{u}_i \leftarrow \mathcal{D}_{g_i, \mathbf{A}, \mathbf{t}, \mathbf{v}},\ \forall i \in \mathbb{Z}_k\right\}$$

is statistically close to the distribution

$$\left\{(\mathbf{A}, \mathbf{t}, \{\mathbf{u}_i\}, \mathbf{v}) \middle| \begin{array}{l} \mathbf{A} \leftarrow \mathcal{R}_q^{\eta_M \times \ell_M};\ \mathbf{t} \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w \\ \mathbf{u}_i \leftarrow \mathcal{D}_{\mathcal{R}^{\ell_R}, \sigma_R} \times \mathcal{D}_{\mathcal{R}^{\ell_\Delta}, \sigma_\Delta} : \mathbf{A} \cdot \mathbf{u}_i \equiv g_i(\mathbf{v}) \cdot \mathbf{t} \bmod q,\ \forall i \in \mathbb{Z}_k \end{array}\right\}.$$

Let SampD *and* $\mathsf{SampPre}$ *output samples following a Discrete Gaussian distribution of appropriate width σ_x given β_x. For any PPT adversary $\mathcal{A}$ against k-M-$\mathsf{ISIS}_{\mathsf{pp}_M}$ there exists a PPT adversary $\mathcal{A}'$ or $\mathcal{A}''$ against k-M-$\mathsf{SIS}_{\mathcal{R}_q,1,\ell_R,\beta_R,\beta_R^*}$ or R-$\mathsf{SIS}_{\mathcal{R}_q,1,\ell_R,\beta_R^*}$ respectively with*

$$\mathsf{Adv}^{\mathsf{k\text{-}r\text{-}sis}}_{\mathcal{R}_q,\ell_R,\beta,\beta_R^*,\mathcal{A}'} + \mathsf{Adv}^{\mathsf{r\text{-}sis}}_{\mathcal{R}_q,\ell_R,\beta_R^*,\mathcal{A}''} \quad \geq \quad \frac{1}{\mathsf{poly}(\lambda)} \cdot \mathsf{Adv}^{\mathsf{k\text{-}m\text{-}isis}}_{\mathsf{pp}_M}.$$

We first state and prove a technical lemma that we will rely on in both proofs. It allows us to argue, using Rényi and statistical distance arguments, that the structured inputs we provide to the k-M-ISIS adversary are sufficiently close to what this adversary expects for it to succeed.

Lemma 3.4.3. *Consider*

$$\mathbf{A} \coloneqq \begin{pmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{r} & \mathbf{b} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{a} & \mathbf{a} \cdot \mathbf{R} \\ \mathbf{r} & \mathbf{b} \end{pmatrix},$$
$$\mathbf{U} \coloneqq \begin{pmatrix} \mathbf{I} & -\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{U}_R \\ \mathbf{W}_\Delta \end{pmatrix} = \begin{pmatrix} \mathbf{U}_R - \mathbf{R} \cdot \mathbf{W}_\Delta \\ \mathbf{W}_\Delta \end{pmatrix},$$

where $\mathbf{a}, \mathbf{r} \leftarrow \mathcal{R}_q^{\ell_R}$, $\mathbf{b} \leftarrow \mathcal{R}_q^{\ell_\Delta}$, $\mathbf{R} \in \mathcal{R}^{\ell_R \times \ell_\Delta}$ with each entry sampled independently from $\mathcal{D}_{\mathcal{R},\sigma}$, $\mathbf{U}_R \in \mathcal{R}^{\ell_R \times k}$ with each entry sampled independently from $\mathcal{D}_{\mathcal{R},\sigma_R}$, $\mathbf{W}_\Delta \in \mathcal{R}^{\ell_\Delta \times k}$ with entry is sampled independently from $\mathcal{D}_{\mathcal{R},\sigma_\Delta}$.

Let $\sigma, \sigma_\Delta > \eta_\epsilon(\mathcal{R}) \in \mathbb{R}$, $\beta_x \geq \sigma_x$ be s.t. $u \sim \mathcal{D}_{\mathcal{R}^\ell,\sigma_x}$ satisfy $\|u\|_\infty \leq \beta_x$ for $x \in \{R, M, \Delta\}$, $\ell_\Delta \coloneqq \ell_M - \ell_R \geq \mathsf{lhl}(\mathcal{R}, 1, q, \beta_\Delta)$, $\sigma_R > \gamma_\mathcal{R} \cdot (\ell_\Delta \cdot n)^{3/2} \cdot \gamma_\mathcal{R} \cdot \sigma_\Delta \cdot \sigma$, $\ell_R \geq \mathsf{lhl}(\mathcal{R}, 1, q, \sigma)$ and $\geq \mathsf{lhl}(\mathcal{R}, 2, q, \beta_R)$.

Let $\mathsf{pp}_{M'}$ *be as in Definition 3.3.2 except that* $\mathbf{A}$ *is sampled as above and* $\mathbf{u}_i$ *are sampled as the columns of* $\mathbf{U}$ *subject to* $\mathbf{A} \cdot \mathbf{U} \equiv \mathbf{G} \cdot \mathbf{t} \bmod q$ *where* $\mathbf{G} \coloneqq \mathcal{M}_{\mathcal{G}}(\mathbf{g})$. *Let* SampD *and* $\mathsf{SampPre}$ *output samples following a Discrete Gaussian distribution of appropriate width* σ_x *given* β_x. *Let* $\mathcal{A}$ *be a* k-M-ISIS *adversary solving instances sampled as in Definition 3.3.2 with non-negligible probability, then* $\mathcal{A}$ *also solves instances with* $\mathsf{pp}_{M'}$ *with non-negligible probability.*

Proof. We argue this by defining a series of hybrid experiments for sampling $(\mathbf{A}, \mathbf{t}, \mathbf{U}, \mathbf{v})$:

Hyb_0**:** The input $(\mathbf{A}_0, \mathbf{t}, \mathbf{U}_0, \mathbf{v})$ is sampled as above.

Hyb_1**:** In this experiment $(\mathbf{A}_1, \mathbf{t}, \mathbf{U}_1, \mathbf{v})$ is sampled such that $\mathbf{U}_1$ is sampled independent of $\mathbf{R}$, i.e. $\mathbf{u}_g \coloneqq (\mathbf{u}_g^{(R)}, \mathbf{u}_g^{(\Delta)})$ where $\mathbf{u}_g^{(R)} \sim \mathcal{D}_{\mathcal{R}^{\ell_R}, \sigma_R}$ and $\mathbf{u}_g^{(\Delta)} \sim \mathcal{D}_{\mathcal{R}^{\ell_\Delta}, \sigma_\Delta}$.

Hyb_2**:** In this experiment $(\mathbf{A}_2, \mathbf{t}, \mathbf{U}_2, \mathbf{v})$ is sampled as in the k-M-ISIS definition.

We first establish the closeness between the distributions Hyb_0 and Hyb_1.

Claim 3.4.1. *The Rényi divergence between* Hyb_0 *and* Hyb_1 *is at most a constant.*

Proof. We first show how we can sample from Hyb_1. Let $\mathbf{R}_1 \leftarrow \mathcal{R}^{\ell_R \times \ell_\Delta}$ be sampled as in Hyb_0. Sample $(\mathbf{X}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(2, \ell_R, q, \mathcal{R}, \beta_R)$, $\mathbf{y} \leftarrow \mathcal{R}_q^{\ell_M}$, write $\mathbf{x}_i$ for the i-th row of $\mathbf{X}$, and set

$$\mathbf{A}_1 \coloneqq \begin{pmatrix} \mathbf{x}_0 & \mathbf{0} \\ \mathbf{x}_1 & \mathbf{y} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{R}_1 \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_0 & \mathbf{x}_0 \cdot \mathbf{R}_1 \\ \mathbf{x}_1 & \mathbf{y} \end{pmatrix}.$$

Note that $\mathbf{x}_0$, $\mathbf{x}_1$, and $\mathbf{y}$ play the roles of $\mathbf{a}$, $\mathbf{r}$, and $\mathbf{b}$ in Hyb_0 respectively. Then, sample $\mathbf{W}_{\Delta,1} \leftarrow \mathcal{D}_{\mathcal{R}^{\ell_\Delta \times k}, \sigma_\Delta}$ and $\mathbf{U}_{R,1} \leftarrow \mathsf{SampPre}\left(\mathsf{td}, \mathbf{G} \cdot \mathbf{t} - \begin{pmatrix} \mathbf{x}_0 \cdot \mathbf{R}_1 \\ \mathbf{y} \end{pmatrix} \mathbf{W}_{\Delta,1}, \beta_R\right)$ so that they satisfy

$$\mathbf{A}_1 \cdot \begin{pmatrix} \mathbf{U}_{R,1} \\ \mathbf{W}_{\Delta,1} \end{pmatrix} \equiv \mathbf{G} \cdot \mathbf{t} \bmod q.$$

We next argue about the closeness of Hyb_0 and Hyb_1. Write

$$\mathbf{U}_0 = ((\mathbf{U}_{R,0} + \mathbf{R}_0 \cdot \mathbf{W}_{\Delta,0})^\intercal \| \mathbf{W}_{\Delta,0}^\intercal)^\intercal.$$

Since $\ell_R \geq \mathsf{lhl}(\mathcal{R}, 2, q, \beta_R)$ and by the properties of $\mathsf{TrapGen}$ we have that $\mathbf{A}_0$ and $\mathbf{A}_1$ are statistically close. We also note that $\mathbf{W}_{\Delta,0}$ and $\mathbf{W}_{\Delta,1}$ are identically distributed. Next, we consider the distribution $\mathcal{D}_{\mathsf{Hyb}_1} \coloneqq \mathcal{D}_{\mathcal{R}^{\ell_R \times k}, \sigma_R}$ of $\mathbf{U}_{R,1}$ and the distribution $\mathcal{D}_{\mathsf{Hyb}_0}$ of $\mathbf{U}_{R,0} + \mathbf{R}_0 \cdot \mathbf{W}_{\Delta,0}$, where we recall that $\mathbf{U}_{R,0} \sim \mathcal{D}_{\mathcal{R}^{\ell_R \times k}, \sigma_R}$, $\mathbf{W}_{\Delta,0} \sim \mathcal{D}_{\mathcal{R}^{\ell_\Delta \times k}, \sigma_\Delta}$ and $\mathbf{R}_0 \sim \mathcal{D}_{\mathcal{R}^{\ell_R \times \ell_\Delta}, \sigma}$. By Proposition 3 $\|\mathbf{W}_{\Delta,0}\| \leq \sqrt{\ell_\Delta \cdot n} \cdot \sigma_\Delta$, each column $\mathbf{r}$ of $\mathbf{R}_0$ satisfies

$\|\mathbf{r}\| \leq \sqrt{\ell_\Delta \cdot n} \cdot \sigma$ and thus $\|\mathbf{R}_0 \cdot \mathbf{W}_{\Delta,0}\|_2 \leq (\ell_\Delta \cdot n)^{3/2} \cdot \gamma_\mathcal{R} \cdot \sigma_\Delta \cdot \sigma$. By Lemma 3.2.2, the Rényi divergence of order $a \in (1, \infty)$ is thus

$$R_a\left(\mathcal{D}_{\mathsf{Hyb}_1} \| \mathcal{D}_{\mathsf{Hyb}_0}\right) \leq \exp\left(a\,\pi \cdot \left((\ell_\Delta \cdot n)^{3/2} \cdot \gamma_\mathcal{R} \cdot \sigma_\Delta \cdot \sigma\right)^2 / (\sigma_R)^2\right).$$

By assumption $\sigma_R > \gamma_\mathcal{R} \cdot (\ell_\Delta \cdot n)^{3/2} \cdot \gamma_\mathcal{R} \cdot \sigma_\Delta \cdot \sigma$ and thus the Rényi divergence $R_a\left(\mathcal{D}_{\mathsf{Hyb}_1} \| \mathcal{D}_{\mathsf{Hyb}_0}\right)$, and hence $R_a\left(\mathsf{Hyb}_1 \| \mathsf{Hyb}_0\right)$, is bounded by a constant. □

Next, let E be the event that the k-M-ISIS adversary is successful when given $(\mathbf{A}, \mathbf{t}, \mathbf{U}, \mathbf{v})$, and denote the probability of this event happening when $(\mathbf{A}, \mathbf{t}, \mathbf{U}, \mathbf{v})$ is sampled from Hyb_1 by $\mathsf{Hyb}_1(E)$. By Lemma 3.2.1 we have that $\mathsf{Hyb}_0(E) \geq \mathsf{Hyb}_1(E)^{a/(a-1)} / R_a(\mathsf{Hyb}_0 \| \mathsf{Hyb}_1)$. Taking any constant $a > 1$ establishes that $(\mathbf{A}_0, \mathbf{t}, \mathbf{U}_0, \mathbf{v})$ sampled from Hyb_0 is sufficiently well distributed for the adversary to succeed if it does for $(\mathbf{A}_1, \mathbf{t}, \mathbf{U}_1, \mathbf{v})$ sampled from Hyb_1.

It remains to show that $\mathsf{Hyb}_1(E)$ and $\mathsf{Hyb}_2(E)$ are (statistically) indistinguishable. For this, we use that $\ell_R \geq \mathsf{lhl}(\mathcal{R}, 1, q, \sigma)$ and the distributions of $\mathbf{a}, \mathbf{b}, \mathbf{r}$ to conclude that $\mathbf{A}_1$ and $\mathbf{A}_2$ are statistically close, which implies that the distributions Hyb_1 and Hyb_2 are statistically close. The statistical indistinguishability between $\mathsf{Hyb}_1(E)$ and $\mathsf{Hyb}_2(E)$ follows. □

Proof of Theorem 3.4.1. Let $(\mathbf{a}, t, \{\, \mathbf{u}_\mathcal{G} \,\}, \mathbf{v}) \in \mathcal{R}_q^{\ell_R} \times \mathcal{R}_q \times \mathcal{R}_q^w \times \mathcal{R}^{\ell_R \times k}$ be a k-R-ISIS instance. Without loss of generality (Remark 5), suppose $t = 1$. Our reduction samples: $(\mathbf{b}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1, \ell_\Delta, q, \mathcal{R}, \beta_\Delta)$, $\mathbf{r} \leftarrow \mathcal{R}_q^{\ell_R}$ and a short matrix $\mathbf{R} \in \mathcal{R}^{\ell_R \times \ell_\Delta}$ where each entry is sampled independently from $\mathcal{D}_{\mathcal{R},\sigma}$.

Let $\mathbf{U} \in \mathcal{R}^{\ell_R \times k} \coloneqq \mathcal{M}_\mathcal{G}(\{\, \mathbf{u}_\mathcal{G} \,\})$. For each $g \in \mathcal{G}$, sample short preimages $\mathbf{w}_g \leftarrow \mathsf{SampPre}(\mathsf{td}, -\langle \mathbf{r}, \mathbf{u}_g \rangle, \beta_\Delta)$. Note that $0 \equiv \langle \mathbf{r}, \mathbf{u}_g \rangle + \langle \mathbf{b}, \mathbf{w}_g \rangle \bmod q$. Let $\mathbf{W} \in \mathcal{R}^{\ell_\Delta \times k} \coloneqq \mathcal{M}_\mathcal{G}(\{\, \mathbf{w}_\mathcal{G} \,\})$ and $\mathbf{G} \in \mathcal{R}_q^{1 \times k} \coloneqq \mathcal{M}_\mathcal{G}(\{\, g(\mathbf{v}) \,\})$. We construct

$$\mathbf{A}' \coloneqq \begin{pmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{r} & \mathbf{b} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{a} & \mathbf{a} \cdot \mathbf{R} \\ \mathbf{r} & \mathbf{b} \end{pmatrix},$$
$$\mathbf{U}' \coloneqq \begin{pmatrix} \mathbf{I} & -\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{U} \\ \mathbf{W} \end{pmatrix} = \begin{pmatrix} \mathbf{U} - \mathbf{R} \cdot \mathbf{W} \\ \mathbf{W} \end{pmatrix}.$$

Without loss of generality (Remark 5), suppose that $\mathbf{t} = (1, 0)^\mathsf{T}$. By construction we have

$$\mathbf{A}' \cdot \mathbf{U}' \equiv \mathbf{G} \cdot \mathbf{t} \bmod q$$

as required. Our reduction runs the k-M-ISIS adversary on $(\mathbf{A}', \mathbf{t}, \mathbf{U}', \mathbf{v})$. When the adversary returns a short preimage $\mathbf{u}^*$ of $s^* \cdot g^*(\mathbf{v}) \cdot \mathbf{t}$ we have

$$\begin{aligned} s^* \cdot g^*(\mathbf{v}) \cdot \mathbf{t} &\equiv \mathbf{A}' \cdot \mathbf{u}^* \bmod q \\ &\equiv \begin{pmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{r} & \mathbf{b} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u}_0^* \\ \mathbf{u}_1^* \end{pmatrix} \bmod q \\ &\equiv \begin{pmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{r} & \mathbf{b} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^* \\ \mathbf{u}_1^* \end{pmatrix} \bmod q \\ s^* \cdot g^*(\mathbf{v}) &\equiv \langle \mathbf{a}, \mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^* \rangle \bmod q, \end{aligned}$$

i.e. $\mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^*$ is a solution for k-R-ISIS. By Proposition 3 the entries of $\mathbf{R}$ are bounded by $\sqrt{n} \cdot \sigma$ with overwhelming probability. Thus, $\|\mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^*\| \leq 2\,\ell_\Delta \cdot \gamma_\mathcal{R} \cdot \sqrt{2\,\pi n} \cdot \sigma \cdot \beta_M^* \leq \beta_R^*$.

Finally, to show that the input $(\mathbf{A}', \mathbf{t}, \mathbf{U}', \mathbf{v})$ to the k-M-ISIS adversary is (sufficiently) well distributed, we apply Lemma 3.4.3. □

of Theorem 3.4.2. Let $(\mathbf{a}, \{\mathbf{u}_i\}) \in \mathcal{R}_q^{\ell_R} \times \mathcal{R}^{\ell_R \times k}$ be a k-R-SIS instance. Our reduction samples: $\mathbf{v} \in (\mathcal{R}_q^\times)^w$, $(\mathbf{b}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1, \ell_\Delta, q, \mathcal{R}, \beta_\Delta)$, $\mathbf{r} \leftarrow \mathcal{R}_q^{\ell_R}$ and a short matrix $\mathbf{R} \in \mathcal{R}^{\ell_R \times \ell_\Delta}$ where each entry is sampled independently from $\mathcal{D}_{\mathcal{R},\sigma}$. Let $\mathbf{U} \in \mathcal{R}^{\ell_R \times k}$ be the matrix where $\mathbf{u}_i$ are the columns. For each $0 \leq i < k$, sample short preimages $\mathbf{w}_{g_i} \leftarrow \mathsf{SampPre}(\mathsf{td}, -\langle \mathbf{r}, \mathbf{u}_i \rangle + g_i(\mathbf{v}), \beta_\Delta)$. Note that $g_i(\mathbf{v}) \equiv \langle \mathbf{r}, \mathbf{u}_{g_i} \rangle + \langle \mathbf{b}, \mathbf{w}_{g_i} \rangle \bmod q$. Let $\mathbf{W} \in \mathcal{R}^{\ell_\Delta \times k} \coloneqq \mathcal{M}_\mathcal{G}(\{\mathbf{w}_\mathcal{G}\})$ and $\mathbf{G} \in \mathcal{R}_q^{1 \times k} \coloneqq \mathcal{M}_\mathcal{G}(\{g(\mathbf{v})\})$. We construct

$$\begin{aligned} \mathbf{A}' &\coloneqq \begin{pmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{r} & \mathbf{b} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{a} & \mathbf{a} \cdot \mathbf{R} \\ \mathbf{r} & \mathbf{b} \end{pmatrix}, \\ \mathbf{U}' &\coloneqq \begin{pmatrix} \mathbf{I} & -\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{U} \\ \mathbf{W} \end{pmatrix} = \begin{pmatrix} \mathbf{U} - \mathbf{R} \cdot \mathbf{W} \\ \mathbf{W} \end{pmatrix}. \end{aligned}$$

Without loss of generality (Remark 5), suppose that $\mathbf{t} = (0, 1)^\mathsf{T}$. By construction we have

$$\mathbf{A}' \cdot \mathbf{U}' \equiv \mathbf{G} \cdot \mathbf{t} \bmod q$$

as required. Our reduction runs the k-M-ISIS adversary on $(\mathbf{A}', \mathbf{t}, \mathbf{U}', \mathbf{v})$. When the adversary returns a short preimage $\mathbf{u}^*$ of $s^* \cdot g^*(\mathbf{v}) \cdot \mathbf{t}$ we have

$$\begin{aligned} s^* \cdot g^*(\mathbf{v}) \cdot \mathbf{t} &\equiv \mathbf{A}' \cdot \mathbf{u}^* \bmod q \\ &\equiv \begin{pmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{r} & \mathbf{b} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{I} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u}_0^* \\ \mathbf{u}_1^* \end{pmatrix} \bmod q \\ &\equiv \begin{pmatrix} \mathbf{a} & \mathbf{0} \\ \mathbf{r} & \mathbf{b} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^* \\ \mathbf{u}_1^* \end{pmatrix} \bmod q \\ 0 &\equiv \langle \mathbf{a}, \mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^* \rangle \bmod q, \end{aligned}$$

i.e. $\mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^*$ is a candidate solution for k-R-SIS. First, we bound its norm. By Proposition 3 the entries of $\mathbf{R}$ are bounded by $\sqrt{n} \cdot \sigma$ with overwhelming probability. Thus, $\|\mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^*\| \leq 2\,\ell_\Delta \cdot \gamma_\mathcal{R} \cdot \sqrt{2\,\pi n} \cdot \sigma \cdot \beta_M^* \leq \beta_R^*$.

Second, we establish that the solution is a valid k-R-SIS solution, i.e. not in the span of the $\mathbf{u}_i$. We distinguish two cases.

$\mathbf{u}_1^* = \mathbf{0}$. In this case we also have $\langle \mathbf{r}, \mathbf{u}_0^* \rangle \equiv g_M^*(\mathbf{v}) \equiv 0 \bmod q$, i.e. $\mathbf{u}_0^*$ is solution to the R-SIS$_{\mathcal{R}_q,\ell_R,\beta_R^*}$ instance $\mathbf{r}$. In other words, if this case happens with non-negligible probability, we could construct a PPT algorithm for R-SIS$_{\mathcal{R}_q,\ell_R,\beta_R^*}$.

$\mathbf{u}_1^* \neq \mathbf{0}$. It remains to be argued that $\mathbf{u}^* \notin \mathcal{K}\text{-span}(\{\,\mathbf{u}_i\,\}_{0\leq i<k})$ with non-negligible probability. First, note that $\mathbf{R}$ is information-theoretically hidden from the k-M-ISIS adversary. Now, suppose the contrary is true, i.e. that we have $\mathbf{u}_0^* + \mathbf{R} \cdot \mathbf{u}_1^* = \sum_{i\in\mathbb{Z}_k} a_i \cdot \mathbf{u}_i$ for some $a_i \in \mathcal{K}$. If this relation holds over $\mathcal{K}$ it must also hold mod 2. By [GPV08, Corollary 2.8], the distribution of $\mathbf{R}$ mod 2 is statistically close to $U(\mathcal{R}_2^{\ell_R \times \ell_\Delta})$ and thus $\mathbf{R} \cdot \mathbf{u}_1^*$ is uniform mod 2. Moreover in the worst case $\mathcal{R}_2$ splits into n copies of $\mathbb{Z}_2$. It suffices to consider only one copy. We thus may ask when $\sum_{i\in\mathbb{Z}_k} a_i \cdot \mathbf{u}_i \equiv \mathbf{R} \cdot \mathbf{u}_i^*$ mod 2 for any $\mathbf{u}_i \in \mathbb{Z}_2^{\ell_R}$ has a solution $a_i \in \{0,1\}^k$. Consider the matrix spanned by $\mathbf{u}_i$ and consider its echelon form. It has at most k pivot positions and thus at least $\ell_R - k$ non-pivot positions. Thus, the probability (over the randomness in $\mathbf{R}$) of satisfying the constraint is $\leq 1/2^{\ell_R - k} \leq 1/2$ since $k < \ell_R$. Thus with probability $> 1/2$ we have $\mathbf{u}^* \notin \mathcal{K}\text{-span}(\{\,\mathbf{u}_i\,\}_{0\leq i<k})$.

Finally, to show that the input $(\mathbf{A}', \mathbf{t}, \mathbf{U}', \mathbf{v})$ to the k-M-ISIS adversary is (sufficiently) well distributed, we apply Lemma 3.4.3. □

$(\mathcal{G}, g^*) \geq (\mathcal{G}, 0)$.

The next lemma shows that solving for any $(\mathcal{G}, g^*)$ is as hard as solving for $(\mathcal{G}, 0)$.

Lemma 3.4.4. *Let the parameters* $\mathsf{pp} = (\mathcal{R}_q, \eta, \ell, w, \mathcal{G}, g^*, \mathcal{D}, \mathcal{T}, \beta, \beta^*)$ *be as in Definition 3.3.2. Furthermore, let* $\beta \leq \beta^*$, $g^* \neq 0$, *and* $\mathcal{D}$ *be such that* $H_\infty(\mathcal{D}_{g^*,\mathbf{A},\mathbf{t},\mathbf{v}}) \geq \lambda$ *for all* $(\mathbf{A}, \mathbf{t}, \mathbf{v})$. *Define* $\hat{\mathsf{pp}} = (\mathcal{R}_q, \eta, \ell, w, \mathcal{G} \cup \{g^*\}, 0, \hat{\mathcal{D}}, \mathcal{T}, \beta, \hat{\beta}^*)$ *where* $\hat{\mathcal{D}} = \mathcal{D} \cup \{\,\mathcal{D}_{g^*,\mathbf{A},\mathbf{t},\mathbf{v}}\,\}_{\mathbf{A},\mathbf{t},\mathbf{v}}$ *and* $\hat{\beta}^* = 2\,\gamma_\mathcal{R} \cdot (\beta^*)^2$. *For any PPT adversary* $\mathcal{A}$ *against* k-M-ISIS$_{\mathsf{pp}}$ *there exists a PPT adversary* $\mathcal{A}'$ *against* k-M-ISIS$_{\hat{\mathsf{pp}}}$ *with*

$$\mathsf{Adv}^{\text{k-m-isis}}_{\hat{\mathsf{pp}},\mathcal{A}'} \quad \geq \quad \frac{1}{\mathsf{poly}(\lambda)} \cdot \mathsf{Adv}^{\text{k-m-isis}}_{\mathsf{pp},\mathcal{A}}.$$

Proof. Upon receiving a k-M-ISIS$_{\hat{\mathsf{pp}}}$ instance $\left(\mathbf{A}, \mathbf{v}, \{\,\mathbf{u}_g\,\}_{g\in\mathcal{G}\cup\{g^*\}}\right)$, $\mathcal{A}'$ runs $\mathcal{A}$ on the k-M-ISIS$_{\mathsf{pp}}$ instance $\left(\mathbf{A}, \mathbf{v}, \{\,\mathbf{u}_g\,\}_{g\in\mathcal{G}}\right)$ and receives from it a vector $(s^*, \mathbf{u}'_{g^*})$.

Our algorithm $\mathcal{A}'$ then outputs $(1, \mathbf{u}'_{g^*} - s^* \cdot \mathbf{u}_{g^*})$. We argue that if $(s^*, \mathbf{u}'_{g^*})$ is a valid solution to the k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ instance then $(1, \mathbf{u}'_{g^*} - s^* \cdot \mathbf{u}_{g^*})$ is a valid solution to the k-M-$\mathsf{ISIS}_{\hat{\mathsf{pp}}}$ instance with non-negligible probability.

Clearly, the k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ instance given to $\mathcal{A}$ is well-distributed. By our assumption on $\mathcal{A}$, with non-negligible probability, it holds that $\mathbf{A} \cdot \mathbf{u}'_{g^*} \equiv s^* \cdot g^*(\mathbf{v}) \cdot \mathbf{t} \bmod q$, $0 < \|s^*\| \leq \beta^*$, and $\left\|\mathbf{u}'_{g^*}\right\| \leq \beta^*$. Since $\mathbf{A} \cdot \mathbf{u}_{g^*} \equiv g^*(\mathbf{v}) \cdot \mathbf{t} \bmod q$, we have $\mathbf{A} \cdot (\mathbf{u}'_{g^*} - s^* \cdot \mathbf{u}_{g^*}) \equiv \mathbf{0} \bmod q$. Furthermore, by our assumption on $\mathcal{D}_{g^*,\mathbf{A},\mathbf{t},\mathbf{v}}$, we have $\|\mathbf{u}_{g^*}\| \leq \beta \leq \beta^*$. We therefore have $\left\|\mathbf{u}'_{g^*} - s^* \cdot \mathbf{u}_{g^*}\right\| \leq 2 \cdot \gamma_{\mathcal{R}} \cdot (\beta^*)^2 = \hat{\beta}^*$. It remains to argue that $\mathbf{u}'_{g^*} - s^* \cdot \mathbf{u}_{g^*} \neq \mathbf{0}$ with non-negligible probability, which is immediate from $H_\infty(\mathcal{D}_{g^*,\mathbf{A},\mathbf{v}}) \geq \lambda$. □

$(\mathcal{G}, g^*) \geq (r \cdot \mathcal{G}, r \cdot g^*)$.

We show that the k-M-ISIS assumption is invariant under multiplication by any non-zero Laurent monomial $r(\mathbf{X})$.

Lemma 3.4.5. *Let the parameters $\mathsf{pp} = (\mathcal{R}_q, \eta, \ell, w, \mathcal{G}, g^*, \mathcal{D}, \mathcal{T}, \beta, \beta^*)$ be as in Definition 3.3.2. Let $r(\mathbf{X}) \in \mathcal{R}(\mathbf{X})$ be a non-zero Laurent monomial and denote $r \cdot \mathcal{G} := \{ r \cdot g : g \in \mathcal{G} \}$. Define $\hat{\mathsf{pp}} = (\mathcal{R}_q, \eta, \ell, w, r \cdot \mathcal{G}, r \cdot g^*, \mathcal{D}, \mathcal{T}, \beta, \beta^*)$. For any PPT adversary $\mathcal{A}$ against k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ there exists a PPT adversary $\mathcal{A}'$ against k-M-$\mathsf{ISIS}_{\hat{\mathsf{pp}}}$ with*

$$\mathsf{Adv}^{\text{k-m-isis}}_{\hat{\mathsf{pp}},\mathcal{A}'} \geq \frac{1}{\mathsf{poly}(\lambda)} \cdot \mathsf{Adv}^{\text{k-m-isis}}_{\mathsf{pp},\mathcal{A}}.$$

Proof. Upon receiving a k-M-$\mathsf{ISIS}_{\hat{\mathsf{pp}}}$ instance $\left(\mathbf{A}, \mathbf{t}, \mathbf{v}, \{ \mathbf{u}_g \}_{g \in \mathcal{G}}\right)$, $\mathcal{A}'$ sets $\mathbf{B} := r(\mathbf{v}) \cdot \mathbf{A}$, which is well-defined since $\mathbf{v} \in (\mathcal{R}_q^\times)^w$. It then runs $\mathcal{A}$ on the k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ instance $\left(\mathbf{B}, \mathbf{t}, \mathbf{v}, \{ \mathbf{u}_g \}_{g \in \mathcal{G}}\right)$ and receives from it a tuple $(s^*, \mathbf{u}_{g^*})$. Our algorithm $\mathcal{A}'$ then outputs $(s^*, \mathbf{u}_{g^*})$. We argue that if $(s^*, \mathbf{u}_{g^*})$ is a valid solution to the k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ instance $\left(\mathbf{B}, \mathbf{t}, \mathbf{v}, \{ \mathbf{u}_g \}_{g \in \mathcal{G}}\right)$, then it is also a valid solution to the k-M-$\mathsf{ISIS}_{\hat{\mathsf{pp}}}$ instance $\left(\mathbf{A}, \mathbf{t}, \mathbf{v}, \{ \mathbf{u}_g \}_{g \in \mathcal{G}}\right)$.

Note that $r(\mathbf{v}) \in \mathcal{R}_q^\times$ and $\mathbf{A}$ is uniformly random over $\mathcal{R}_q^{\eta \times \ell}$. Therefore $\mathbf{A}$ is also uniformly random over $\mathcal{R}_q^{\eta \times \ell}$. Next, note that $\mathbf{B} \cdot \mathbf{u}_g = r(\mathbf{v}) \cdot \mathbf{A} \cdot \mathbf{u}_g \equiv (r \cdot g)(\mathbf{v}) \cdot \mathbf{t} \bmod q$. The k-M-$\mathsf{ISIS}_{\mathsf{pp}}$ instance given to $\mathcal{A}$ is therefore well-distributed.

By our assumption on $\mathcal{A}$, with non-negligible probability, it holds that $\mathbf{B} \cdot \mathbf{u}_{g^*} \equiv s^* \cdot g^*(\mathbf{v}) \cdot \mathbf{t} \bmod q$, $0 < \|s^*\| \leq \beta^*$, $\|\mathbf{u}_{g^*}\| \leq \beta^*$, and $(g^*, \mathbf{u}_{g^*}) \neq (0, \mathbf{0})$. The first equation implies

$$\begin{aligned} s^* \cdot (r \cdot g^*)(\mathbf{v}) \cdot \mathbf{t} &\equiv r(\mathbf{v}) \cdot \mathbf{A} \cdot \mathbf{u}_{g^*} \bmod q \\ &\equiv \mathbf{B} \cdot \mathbf{u}_{g^*} \bmod q. \end{aligned}$$

□

3.4.1 Attacks

Our first attack simply solves M-ISIS (more precisely ISIS). It thus simply ignores the algebraic dependencies among the $\{ g(\cdot) \}_{g\in\mathcal{G}}$. Our further attacks attempt to find short linear combinations among the $\{ g(\mathbf{v}) \}_{g\in\mathcal{G}}$.

Direct SIS Attack.

First, we can reduce the problem of finding $\mathbf{u}_{g^*}$ s.t. $\mathbf{A}\cdot\mathbf{u}_{g^*} \equiv s^*\cdot g^*(\mathbf{v})\cdot\mathbf{t} \bmod q$ to finding a $\mathbf{A}'\cdot\mathbf{u}'_{g^*} \equiv 0 \bmod q$ with $\mathbf{A}' \coloneqq (\mathbf{A}, -g^*(\mathbf{v})\cdot\mathbf{t})$. Then the last entry of $\mathbf{u}'_{g^*}$ becomes s^*. The analysis here is completely standard.

We will write this as $\mathbf{A}\cdot\mathbf{u} \equiv 0 \bmod q$ with $\mathbf{A}\in\mathbb{Z}^{n\cdot\eta\times(n\cdot\eta\cdot(\ell+1))}$. This task is equivalent to finding a short vector in $\Lambda(\mathbf{L})$ with $\mathbf{A}\cdot\mathbf{L}\equiv 0 \bmod q$ and $\mathbf{L}\in\mathbb{Z}_q^{(n\cdot\eta\cdot(\ell+1))\times(n\cdot\eta\cdot\ell)}$. Thus, we are trying to find a short vector in a $d \le n\cdot\eta\cdot(\ell+1)$ dimensional lattice with volume $\mathrm{Vol}(\Lambda) = q^{n\cdot\eta}$. Our problem formulation is for the infinity norm but lattice reduction naturally considers the ℓ_2 norm. We thus consider it a win if lattice reduction finds a vector of norm $\sqrt{d}\cdot\beta^*$, which is generous to the attacker. That is, we are trying to establish the root-Hermite factor δ s.t.

$$\sqrt{d}\cdot\beta^* \approx \delta^{d-1}\cdot\mathrm{Vol}(\Lambda)^{1/d}.$$

The minimum of the right hand side attained at $d \approx \sqrt{n\cdot\eta\cdot\log q/\log\delta}$.[7] Overall, we obtain a vector of norm $2^{2\cdot\sqrt{n\cdot\eta\cdot\log(\delta)\cdot\log(q)}-\log(\delta)}$.

A Solution in $\mathsf{Span}_{\mathcal{R}}(\{ \mathbf{u}_g \}_{g\in\mathcal{G}})$.

We note that $\mathbf{v}\leftarrow\mathcal{R}_q^w$ is critical for security. If all v_i are small then e.g. $v_0/v_1\cdot\mathbf{t} \equiv \mathbf{A}\cdot\mathbf{u}_{X_0/X_1}$ and $v_2/v_1\cdot\mathbf{t} \equiv \mathbf{A}\cdot\mathbf{u}_{X_2/X_1}$ (which corresponds to the form of $\mathcal{G}$ which we will consider below) allows to compute $\mathbf{A}\cdot\left(v_2\cdot\mathbf{u}_{X_0/X_1} - v_0\cdot\mathbf{u}_{X_2/X_1}\right) \equiv 0 \bmod q$. If $k > \ell$ linearly independent such preimages of zero can be constructed then this constitutes a trapdoor for $\mathbf{A}$ and solves k-M-ISIS.

More generally and for $\mathbf{v}\leftarrow\mathcal{R}_q^w$, we may attempt to find a short $\mathbf{z} = (z_{g_0},\ldots,z_{g_{k-1}})$ s.t.

$$\langle (g_0(\mathbf{v}),\ldots,g_{k-1}(\mathbf{v})), \mathbf{z}\rangle \equiv s^*\cdot g^*(\mathbf{v}) \bmod q,$$

for $g_i\in\mathcal{G}$. We then compute $\mathbf{u}_{g^*} = \sum_{g\in\mathcal{G}} z_g\cdot\mathbf{u}_g$ which gives

$$\mathbf{A}\cdot\mathbf{u}_{g^*} = \mathbf{A}\cdot\left(\sum_{g\in\mathcal{G}} z_g\cdot\mathbf{u}_g\right) = \sum_{g\in\mathcal{G}} z_g\cdot\mathbf{A}\cdot\mathbf{u}_i = \sum_{g\in\mathcal{G}} z_g\cdot g(\mathbf{v})\cdot\mathbf{t} = s^*\cdot g^*(\mathbf{v})\cdot\mathbf{t} \bmod q.$$

[7] The minimum is $d = \sqrt{n\log q/\log\delta}$ for $\beta_\ell = \delta^d\cdot\mathrm{Vol}(\Lambda)^{1/d}$ which is what the literature typically considers. However, normalising δ by $d-1$ instead of d makes sense from the analysis of lattice algorithms. Note that $d \ge 1000$ and $\delta < 1.02$ so that discrepancy is tiny.

Write $\mathbf{G} = [\mathrm{rot}(g_0(\mathbf{v}))|\ldots|\mathrm{rot}(g_{k-1}(\mathbf{v}))|\mathrm{rot}(g^*(\mathbf{v}))] \in \mathbb{Z}_q^{n\times(n\cdot(k+1))}$. As above, we cost finding a short vector in $\Lambda(\mathbf{W})$ where $\mathbf{G}\cdot\mathbf{W} \equiv 0 \bmod q$. The analysis proceeds exactly as above.

One the one hand, the final solution will have a larger expected norm $\leq \sqrt{k}\cdot\gamma_R\cdot\max_{g\in\mathcal{G}}(\beta_g)\cdot\beta_{\mathbf{z}}$ when $\|\mathbf{z}\| \leq \beta_{\mathbf{z}}$: we are adding up k terms, each being the product of two elements, and consider the expected norm. On the other hand, note that this attack is independent of η. This implies that while k-M-ISIS is at least as hard as k-R-ISIS it cannot, in general, be strictly harder.

A Solution in $\mathsf{Span}_{\mathcal{R}_q}(\{\mathbf{u}_g\}_{g\in\mathcal{G}})$.

We can generalise the previous approach to finding any, i.e. not necessarily short, $\mathbf{z} \in \mathcal{R}_q^k$ s.t.

$$\sum z_i\cdot g_i(\mathbf{v}) \equiv s^*\cdot g^*(\mathbf{v}) \bmod q \quad \text{and} \quad \sum z_i\cdot\mathbf{u}_{g_i} = \mathbf{u}_{\mathbf{z}} \text{ with } \|\mathbf{u}_{\mathbf{z}}\| \leq \beta^*.$$

$$\text{Write } \mathbf{G} = \begin{pmatrix}\mathrm{rot}(g_0(\mathbf{v})) & \cdots & \mathrm{rot}(g_{k-1}(\mathbf{v}))\end{pmatrix} \in \mathbb{Z}_q^{n\times(n\cdot k)}$$

$$\mathbf{U} = \begin{pmatrix}\mathrm{rot}\left((\mathbf{u}_{g_0})_0\right) & \cdots & \mathrm{rot}\left((\mathbf{u}_{g_{k-1}})_0\right)\\ & \ddots & \\ \mathrm{rot}\left((\mathbf{u}_{g_0})_{\ell-1}\right) & \cdots & \mathrm{rot}\left((\mathbf{u}_{g_{k-1}})_{\ell-1}\right)\end{pmatrix} \in \mathbb{Z}^{(n\cdot\ell)\times(n\cdot k)}$$

and consider the lattice spanned by the columns of

$$\mathbf{S} \coloneqq \begin{pmatrix}\tau & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{g}^* & q\cdot\mathbf{I}_n & \mathbf{0} & \mathbf{G}\\ \mathbf{0} & \mathbf{0} & q\cdot\mathbf{I}_{n\cdot\ell} & \mathbf{U}\end{pmatrix}$$

where τ is some "embedding factor" optimised by the solving algorithm (the reader may simply assume $\tau = 1$.) Then $\Lambda(\mathbf{S})$ contains a short vector $(-\tau\cdot s^*, \mathbf{0}^\mathsf{T}, \mathbf{u}_{\mathbf{z}}^\mathsf{T})^\mathsf{T}$. Computing the column Hermite normal form of $\mathbf{S}$ produces a basis in $\mathbb{Z}^{d\times d}$ with $d \coloneqq \ell\cdot(n+1)$. Assuming full row rank of $[\mathbf{G}^\mathsf{T}, \mathbf{U}^\mathsf{T}] \bmod q$, the determinant of $\Lambda(\mathbf{S})$ is q^t with $t \coloneqq (\ell - \min(k,\ell) + 1)\cdot n$.

Thus, by the Gaussian heuristic, i.e. assume the lattice generated behaves like a random lattice, we expect a shortest vector to have norm $\approx \sqrt{d/2\,\pi\,e}\cdot q^{t/d}$ and lattice reduction with root Hermite factor δ to find a vector of norm $\delta^{d-1}\cdot q^{t/d}$. This is minimised when $t = \varphi(m)$, i.e. when $k = \ell$.

3.4.2 Knowledge Assumptions.

Finally, we consider the knowledge assumptions.

On the one hand, we evaluate these attack strategies with respect to the knowledge assumption. An adversary that succeeds with the direct SIS strategy breaks our knowledge

assumption while also breaking the M-SIS assumption. The second approach – finding a solution in $\mathsf{Span}_{\mathcal{R}}(\{\mathbf{u}_g\}_{g\in\mathcal{G}})$ immediately implies the extractor in Definition 3.3.5 by computing x_g directly. The third attack approach – finding a solution in $\mathsf{Span}_{\mathcal{R}_q}(\{\mathbf{u}_g\}_{g\in\mathcal{G}})$ – initially seems most promising to invalidate our knowledge assumption by generalising the attack to find large x_g such that $\mathbf{u}^* := \sum_{g\in\mathcal{G}} x_g \cdot \mathbf{u}_g$ is small. While finding such x_g given $\mathbf{u}^*$ and $\mathbf{u}_g$ is easy, finding a suitable target $\mathbf{u}^*$, i.e. one satisfying $c\cdot\mathbf{t} \equiv \mathbf{A}\cdot\mathbf{u}^* \bmod q$, seems hard, as outlined above.

On the other hand, we highlight a gap between the knowledge and "plain" k-R-SIS assumption pair and the knowledge k-M-SIS and plain k-M-SIS pair. For k-R-SIS, we can wlog pick $t = 1$ for the plain version but must pick $t \neq 1$ such that $1/|\langle t\rangle| = \mathsf{negl}(\lambda)$ and $|\langle t\rangle|/|\mathcal{R}_q| = \mathsf{negl}(\lambda)$ for the knowledge version. For k-M-SIS, we may pick $\mathbf{t} = (1, 0, \ldots, 0)^T$ in both cases.

This distinction is not just aesthetic. For plain k-R-SIS, the attack strategies we are aware of rely on finding short vectors in some modules of rank > 1. In contrast, in what follows, we sketch an attack on knowledge k-R-SIS which relies on finding short vectors in ideals rather than in modules. Consider $g^* \equiv 0$ and consider $\mathcal{I} := \{s \in \mathcal{R} \mid s\cdot t \equiv 0 \bmod q\}$. Note that $\mathcal{I}$ is an ideal in $\mathcal{R}$ and that finding a sufficiently short element $s' \neq 0$ in $\mathcal{I}$ is a solution for k-R-SIS with target $g^* \equiv 0$. Pick any of the provided $\mathbf{u}_i$ and return $s'\cdot\mathbf{u}_i$. Since it holds that $g_i(\mathbf{v})\cdot t \equiv \langle\mathbf{a}, \mathbf{u}_i\rangle \bmod q$ and $s'\cdot t \equiv 0 \bmod q$ we have that $0 \equiv \langle\mathbf{a}, s'\cdot\mathbf{u}_i\rangle \bmod q$. Finding such an s' efficiently breaks the knowledge assumption since w.h.p. $s'\cdot g_i(\mathbf{v}) \neq 0 \bmod q$. This motivates our restriction in Definition 4.3.3.

To understand what choices of $(\mathcal{R}_q, \mathcal{T})$ may provide secure instantiations, we first note that a series of works [Ber14, CGS14, CDPR16, CDW17, PHS19, DPW19] reports quantum algorithms for finding short vectors in ideal latices that beat known algorithms for general lattices. In particular, finding vectors of norm $2^{\tilde{\mathcal{O}}(\sqrt{m})}$ in an ideal lattice of dimension m can be done in quantum polynomial time. Thus, when $\alpha^* \approx 2^{\tilde{\mathcal{O}}(\sqrt{m})}$ then knowledge k-R-SIS is easy for a quantum adversary.[8]

Moreover, for some choices of ideals in a power-of-two cyclotomic ring $\mathcal{R}$ it has been shown in [PXWC21] that finding short vectors is easy. In our case, by construction, the ideals $\mathcal{S} \subseteq \mathcal{R}$ have algebraic norm $N(\mathcal{S}) = q^i$ for some integer i. The headline result of [PXWC21] thus implies that knowledge k-R-SIS is easy when m is a power of two, $q \equiv \pm 3 \bmod 8$ and $\alpha^* \geq \sqrt{q/(m/2)}$.[9] On the other hand, note that the attack does not apply, for example, when $\mathcal{R}_q$ splits completely into $\phi(m)$ fields, e.g. when $q \equiv 1 \bmod m$ where m is a power of two. Then, sampling t as mentioned in the footnote to Definition 4.3.3, i.e. picking half the CRT components zero and the other half non-zero, produces ideals where, to the best of our knowledge, (approximate) ideal-SVP is not easier than the general case discussed above.

Yet, given that the status of ideal-SVP in $\mathcal{R}_q$ is still in flux, the above highlights that knowledge k-R-SIS is a more risky assumption than (knowledge) k-M-SIS or plain

[8] These quantum improvements may only matter, though, in practice for large values of m [DPW19].
[9] We consider the infinity norm but [PXWC21] considers the Euclidean norm.

k-R-ISIS. In particular, we note that $\mathcal{I} \coloneqq \{s \in \mathcal{R} \mid s \cdot t \equiv 0 \bmod q\} = \{0\}$ for plain k-R-ISIS when $t = 1$, and $\mathcal{I} \coloneqq \{s \in \mathcal{R} \mid s \cdot \mathbf{t} \equiv \mathbf{0} \bmod q\} = \{0\}$ for both plain and knowledge k-M-ISIS when $\mathbf{t} \coloneqq (1, 0, \ldots, 0)^T$, i.e. this line of attack is ruled out.

3.5 Compact Extractable Vector Commitments

We construct compact extractable vector commitments with openings to constant-degree multivariate polynomial maps from the knowledge k-M-ISIS assumption.

3.5.1 Construction

A formal description of our VC construction is in Figure 3.1 where important parameters and shorthands are listed and explained in Table 3.1.

The public parameters consists of a k-M-ISIS instance $(\mathbf{A}_0, \mathbf{t}_0, \mathbf{v}, (\mathbf{u}_{0,g})_{g \in \mathcal{G}_0})$ over $\mathcal{R}_q$, a correlated k-M-ISIS of knowledge instance $(\mathbf{A}_1, \mathbf{t}_1, \mathbf{v}, (\mathbf{u}_{1,g})_{g \in \mathcal{G}_1})$ over $\mathcal{R}_q$ sharing the same $\mathbf{v}$ as the k-M-ISIS instance, and a R-SIS instance $\mathbf{h}$ over $\mathcal{R}_p$, where p is short relative to q. Intuitively, the k-M-ISIS instance is for weak binding, the knowledge k-M-ISIS instance is for upgrading weak binding to extractability, and the R-SIS instance is for compactness. The commitment c to a vector $\mathbf{x}$ is simply $c \coloneqq \langle \mathbf{v}, \mathbf{x} \rangle \bmod q$.

We next explain the opening and verification mechanism. Suppose for the moment that $f(\mathbf{z}, \cdot)$ is a single-output polynomial, i.e. $t = 1$. Consider the commitment c of $\mathbf{x}$ and the evaluation of $f(\mathbf{z}, \cdot)$ at $(v_0^{-1} \cdot c, \ldots, v_w^{-1} \cdot c)$ as polynomials in $\mathbf{v}$. The value $f(\mathbf{z}, \mathbf{x})$ is encoded as the constant term in the evaluation polynomial. To open the commitment c of $\mathbf{x}$ to a function $f(\mathbf{z}, \cdot)$, the committer computes the coefficient of each non-zero Laurent monomial $g \in \mathcal{G}_0$ in the evaluation polynomial, and use these coefficients to compute a linear combination of $(\mathbf{u}_{0,g})_{g \in \mathcal{G}_0}$ to produce $\mathbf{u}_0$. In general, for $t \geq 1$, the committer further compresses the multiple instances of $\mathbf{u}_0$ into a single one using a linear combination with coefficients given by $\mathbf{h}$. To enable extraction (in the security proof), the committer also provides $\mathbf{u}_1$ which is a linear combination of $(\mathbf{u}_{1,g})_{g \in \mathcal{G}_1}$ using $\mathbf{x}$ as coefficients. Given the above, the meaning behind the verification algorithm is immediate.

Finally, we explain the choice of p and q in Table 3.1. First, p is chosen such that the element $f(\mathbf{z}, \mathbf{x}) - y(\mathbf{z})$ is considered short (in the context of R-SIS problems) relative to p for all $f \in \mathcal{F}_{s,w,t}$, $y \in \mathcal{Y}_{s,t}$, $\mathbf{z} \in \mathcal{X}^s$, and $\mathbf{x} \in \mathcal{X}^w$. By some routine calculations, we can see that for such choice of $(f, \mathbf{z}, \mathbf{x}, y)$, we have $\|f(\mathbf{z}, \mathbf{x}) - y(\mathbf{z})\| \leq (s + w + d)^d \cdot \alpha^{d+1} \cdot \gamma_{\mathcal{R}}^d$. More generally, for arbitrary $\mathbf{x} \in \mathcal{R}^w$, we get $\|f(\mathbf{z}, \mathbf{x}) - y(\mathbf{z})\| \leq (s + w + d)^d \cdot \alpha \cdot \|\mathbf{x}\|^d \cdot \gamma_{\mathcal{R}}^d$. As mentioned in Section 3.2.4, a standard choice for R-SIS problems over $\mathcal{R}_p$ is for p to be at least $n \log n$ times the norm bound; we thus simply pick this. Similarly, q is chosen such that δ_0 and δ_1 are both considered short relative to q, concretely by setting q to be $n \log n$ times the maximum among them.[10]

[10] In practice the gap may be smaller or larger and when picking parameters we optimise over these gaps.

Table 3.1: Parameters and shorthands with λ as security parameter.

$s \in \mathbb{N}$		Dimension of public input $\mathbf{z}$
$w \in \mathbb{N}$		Dimension of $\mathbf{v}$ and secret input $\mathbf{x}$
$t \in \mathbb{N}$		Number of outputs
$d \in \mathbb{N}$	$O(1)$	Degree of polynomial maps
$n \in \mathbb{N}$	$\mathsf{poly}(\lambda)$	Degree of $\mathcal{R}$
$\alpha \in \mathbb{R}$	$\mathsf{poly}(\lambda)$	Norm bound for f and $\mathbf{x}$
$\beta \in \mathbb{R}$	$\mathsf{poly}(\lambda)$	Norm bound for public preimages
$\delta_i \in \mathbb{R}$	$\mathsf{poly}(\lambda, s, w, t)$ (Theorem 3.5.1)	Norm bound for honestly generated opening proof $\mathbf{u}_i$
$\delta'_0 \in \mathbb{R}$	$\mathsf{poly}(\lambda, s, w, t)$ (Theorem 3.5.3)	Norm bound for opening proof $\mathbf{u}'_0$ generated by knowledge extractor
$\delta_p \in \mathbb{R}$	$(s+w+d)^d \cdot \alpha^{d+1} \cdot \gamma_{\mathcal{R}}^d$	Norm bound of evaluation of a degree-d $(s+w)$-variate polynomial with coefficients of norm bounded by α at a point of norm bounded by α
$\delta'_p \in \mathbb{R}$	$(s+w+d)^d \cdot \alpha^{d+1} \cdot (w \cdot \beta \cdot \gamma_{\mathcal{R}}^2)^d$	Norm bound of evaluation of a degree-d $(s+w)$-variate polynomial with coefficients of norm bounded by α at a point of norm bounded by $w \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}$
$p \in \mathbb{N}$	$\geq \delta_p \cdot n \cdot \log n$	Moduli for $\mathcal{R}_p$
$q \in \mathbb{N}$	$\geq \max\{\delta'_0, \delta_1\} \cdot n \log n$	Moduli for $\mathcal{R}_q$
$\eta_i \in \mathbb{N}$	$O(1)$	Number of rows of $\mathbf{A}_i$
$\ell_i \in \mathbb{N}$	$\geq \mathsf{lhl}(\mathcal{R}, \eta_i, q, \beta)$	Number of columns of $\mathbf{A}_i$
$\mathcal{X} \subseteq \mathcal{R}$	$\{ x \in \mathcal{R} : \|x\| \leq \alpha \}$	$\mathcal{R}$ elements with norm bound α
$\mathcal{F}_{s,w,t}$		Degree-d $(s+w)$-variate t-output homogeneous polynomial maps over $\mathcal{X}$
$\mathcal{Y}_{s,t}$		s-variate t-output polynomial maps over $\mathcal{X}$
$\mathcal{E}_k \subseteq \mathbb{N}_0^w$	$\{ \mathbf{e} \in \mathbb{N}_0^w : \|\mathbf{e}\|_1 = k \}$	Non-negative integer vectors of 1-norm k, for $k \in [d]$
$\mathcal{G}_0 \subseteq \mathcal{R}(\mathbf{X})$	$\bigcup_{k=1}^{d} \{ \mathbf{X}^{\mathbf{e}'-\mathbf{e}} : \mathbf{e}' \neq \mathbf{e} \in \mathcal{E}_k \}$	Laurent monomials expressible as ratios of distinct degree-k monomials, for $k \in [d]$
$\mathcal{G}_1 \subseteq \mathcal{R}(\mathbf{X})$	$\{ X_i : i \in \mathbb{Z}_w \}$	Degree-1 monomials
$\binom{k}{\mathbf{e}}$	$\binom{k}{e_0, \ldots, e_{w-1}}$	Multinomial coefficient, for $\mathbf{e} \in \mathcal{E}_k$ and $k \in [d]$
$\mathcal{T}_i$		Subset of $\mathcal{R}_q^{\eta_i}$ (Definition 3.3.2)
$f_{i,\mathbf{e}}$		For $f(\mathbf{Z}, \mathbf{X}) \in \mathcal{F}_{s,w,t}$, $f_{i,\mathbf{e}}(\mathbf{Z})$ is the coefficient of the monomial $\mathbf{X}^{\mathbf{e}}$ of the i-th output

Remark 8 (Updating Commitments and Opening Proofs)**.** *We discuss the cost of updating a commitment of* $\mathbf{x}$ *to that of* $\mathbf{x}'$*, and an opening proof for* $f(\mathbf{z}, \mathbf{x})$ *to that of* $f'(\mathbf{z}', \mathbf{x}')$*, omitting fixed* $\mathsf{poly}(\lambda)$ *factors. Due to the linearity of the commitment* $c = \langle \mathbf{v}, \mathbf{x} \rangle \bmod q$ *and opening proof component* $\mathbf{u}_1 = \sum_{i \in \mathbb{Z}_w} x_i \cdot \mathbf{u}_{1,X_i}$ *in the committed vector* $\mathbf{x}$*, they can be updated for a new committed vector* $\mathbf{x}'$ *easily by adding* $\langle \mathbf{v}, \mathbf{x}' - \mathbf{x} \rangle \bmod q$ *and* $\sum_{i \in \mathbb{Z}_w} (x'_i - x_i) \cdot \mathbf{u}_{1,X_i}$ *respectively. The computation complexity of the update is* $O(\Delta)$*, where* Δ *is the Hamming distance between* $\mathbf{x}$ *and* $\mathbf{x}'$*. Updating the* $\mathbf{u}_{0,\mathbf{e}}$ *terms is more computationally expensive due to its non-linearity in* $\mathbf{x}$*. The cost of computing the*

$\mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$

$\mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w$
$\mathbf{h} \leftarrow \mathcal{R}_p^t$
for $i \in \{0,1\}$ **do**
  $(\mathbf{A}_i, \mathsf{td}_i) \leftarrow \mathsf{TrapGen}(1^{\eta_i}, 1^{\ell_i}, q, \mathcal{R}, \beta)$
  $\mathbf{t}_i \leftarrow \mathcal{T}_i$
  $\mathbf{u}_{i,g} \leftarrow \mathsf{SampPre}(\mathsf{td}_i, g(\mathbf{v}) \cdot \mathbf{t}_i, \beta),\ \forall g \in \mathcal{G}_i$
return $\mathsf{pp} := \begin{pmatrix} \mathbf{A}_0, & \mathbf{t}_0, & (\mathbf{u}_{0,g})_{g\in\mathcal{G}_0}, \\ \mathbf{A}_1, & \mathbf{t}_1, & (\mathbf{u}_{1,g})_{g\in\mathcal{G}_1}, \\ & \mathbf{v}, & \mathbf{h} \end{pmatrix}$

$\mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$

$\mathbf{u}_0 := \sum_{i\in\mathbb{Z}_t} \sum_{k=1}^{d} \sum_{\mathbf{e}\in\mathcal{E}_k} h_i \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{u}_{0,\mathbf{e}}$
return $\pi := (\mathbf{u}_0, \mathbf{u}_1)$

$\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, c, \pi)$

$b_0 := \left(\mathbf{A}_0 \cdot \mathbf{u}_0 \stackrel{?}{\equiv} \hat{f}_y(\mathbf{z}, c) \cdot \mathbf{t}_0 \bmod q\right)$
$b_1 := \left(\mathbf{A}_1 \cdot \mathbf{u}_1 \stackrel{?}{\equiv} c \cdot \mathbf{t}_1 \bmod q\right)$
$b_2 := \left(\|\mathbf{u}_0\| \stackrel{?}{\leq} \delta_0\right); b_3 := \left(\|\mathbf{u}_1\| \stackrel{?}{\leq} \delta_1\right)$
return $b_0 \wedge b_1 \wedge b_2 \wedge b_3$

$\mathsf{Com}(\mathsf{pp}, \mathbf{x})$

$c := \langle \mathbf{v}, \mathbf{x} \rangle \bmod q; \quad \mathbf{u}_1 := \sum_{X_i \in \mathcal{G}_1} x_i \cdot \mathbf{u}_{1,X_i}$

for $\mathbf{e} \in \bigcup_{k\in[d]} \mathcal{E}_k$ **do** $\mathbf{u}_{0,\mathbf{e}} := d! \cdot \sum_{\mathbf{e}'\in\mathcal{E}_k\setminus\{\mathbf{e}\}} \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{u}_{0,\mathbf{X}^{\mathbf{e}'-\mathbf{e}}}$

$\mathsf{aux} := \left((\mathbf{u}_{0,\mathbf{e}})_{\mathbf{e}\in\bigcup_{k\in[d]}\mathcal{E}_k}, \mathbf{u}_1\right)$
return (c, aux)

$\mathsf{PreVerify}(\mathsf{pp}, (f, y))$

if $(f, y) \notin \mathcal{F}_{s,w,t} \times \mathcal{Y}_{s,t}$ **then return** $\perp$
$\hat{f}_y(\mathbf{Z}, C) := d! \cdot \left(\sum_{i\in\mathbb{Z}_t} h_i \cdot \left(\sum_{k=1}^{d} \sum_{\mathbf{e}\in\mathcal{E}_k} \binom{k}{\mathbf{e}}^{-1} \cdot f_{i,\mathbf{e}}(\mathbf{Z}) \cdot \mathbf{v}^{-\mathbf{e}} \cdot C^k - y_i(\mathbf{Z})\right)\right)$
$\mathsf{pp}_{f,y} := \left(\mathbf{A}_0, \mathbf{t}_0, \mathbf{A}_1, \mathbf{t}_1, \hat{f}_y\right)$
return $\mathsf{pp}_{f,y}$

Figure 3.1: Our VC Construction.

difference term for $\mathbf{u}_{0,\mathbf{e}}$ *is linear in* $\binom{w}{k} - \binom{w-\Delta}{k} = O(\Delta^k)$ *for each* $\mathbf{e} \in \mathcal{E}_k$ *and each* $k \in [d]$. *The total work needed for updating* $\{\mathbf{u}_{0,\mathbf{e}}\}_{\mathbf{e}\in\mathcal{E}_k,k\in[d]}$ *is thus* $O(w^d \cdot \Delta^d)$. *For fixed* $\mathbf{x}$ *and hence fixed* $\{\mathbf{u}_{0,\mathbf{e}}\}_{\mathbf{e}\in\mathcal{E}_k,k\in[d]}$, *updating* $\mathbf{u}_0$ *by the same method costs computation linear in*

the Hamming distance between the coefficient vector of $f(\mathbf{z},\cdot)$ and that of $f'(\mathbf{z}',\cdot)$.

We show that our VC construction is correct, extractable under a knowledge k-M-ISIS assumption, and compact.

Theorem 3.5.1. *For $d = O(1)$, $\ell_0 := \ell_1 := \mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$,*

$$\delta_0 \geq 2 \cdot p \cdot t \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{2d+1} \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2} \quad \text{and} \quad \delta_1 \geq w \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}, \tag{3.1}$$

our VC construction in Figure 3.1 is correct.

Proof. The multinomial theorem states that $(z_0 + \cdots + z_{w-1})^k = \sum_{\mathbf{e}\in\mathcal{E}_k} \binom{k}{\mathbf{e}} \cdot \mathbf{z}^{\mathbf{e}}$. Let $(c, \mathsf{aux}) = \mathsf{Com}(\mathsf{pp}, \mathbf{x})$ so that $c = \langle \mathbf{v}, \mathbf{x} \rangle = v_0 \cdot x_0 + \cdots + v_{w-1} \cdot x_{w-1}$. Substituting $\mathbf{z} = (v_0 \cdot x_0, \ldots, v_{w-1} \cdot x_{w-1})$ we have $c^k = \sum_{\mathbf{e}\in\mathcal{E}_k} \binom{k}{\mathbf{e}} \cdot \mathbf{v}^{\mathbf{e}} \cdot \mathbf{x}^{\mathbf{e}}$.

Fix any $f \in \mathcal{F}_{s,w,t}$ and any $y \in \mathcal{Y}_{s,t}$. Write $f(\mathbf{Z}, \mathbf{X}) = (\sum_{k=1}^{d} \sum_{\mathbf{e}\in\mathcal{E}_k} f_{i,\mathbf{e}}(\mathbf{Z}) \cdot \mathbf{X}^{\mathbf{e}})_{i\in\mathbb{Z}_t}$ and $y(\mathbf{Z}) = (y_i(\mathbf{Z}))_{i\in\mathbb{Z}_t}$. For $i \in \mathbb{Z}_t$, let

$$\bar{f}_{i,k}(\mathbf{Z}, C) := \sum_{\mathbf{e}\in\mathcal{E}_k} \binom{k}{\mathbf{e}}^{-1} \cdot f_{i,\mathbf{e}}(\mathbf{Z}) \cdot \mathbf{v}^{-\mathbf{e}} \cdot C^k$$

so that $\hat{f}_y(\mathbf{Z}, C) = \sum_{i\in\mathbb{Z}_t} h_i \cdot d! \cdot (\sum_{k=1}^{d} \hat{f}_{i,k}(\mathbf{Z}, C) - y_i(\mathbf{Z}))$.

For any $(\mathbf{z}, \mathbf{x}) \in \mathcal{X}^s \times \mathcal{X}^w$ and any $(c, \mathsf{aux}) \in \mathsf{Com}(\mathsf{pp}, \mathbf{x})$, we observe that

$$\begin{aligned}
\bar{f}_{i,k}(\mathbf{z}, c) &= \sum_{\mathbf{e}\in\mathcal{E}_k} \binom{k}{\mathbf{e}}^{-1} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{v}^{-\mathbf{e}} \cdot c^k \\
&= \left(\sum_{\mathbf{e}\in\mathcal{E}_k} \binom{k}{\mathbf{e}}^{-1} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{v}^{-\mathbf{e}} \right) \cdot \left(\sum_{\mathbf{e}\in\mathcal{E}_k} \binom{k}{\mathbf{e}} \cdot \mathbf{x}^{\mathbf{e}} \cdot \mathbf{v}^{\mathbf{e}} \right) \\
&= \sum_{\mathbf{e},\mathbf{e}'\in\mathcal{E}_k} \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{v}^{\mathbf{e}'-\mathbf{e}} \\
&= \sum_{\mathbf{e}\in\mathcal{E}_k} f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}} + \sum_{\mathbf{e},\mathbf{e}'\in\mathcal{E}_k : \mathbf{e}\neq\mathbf{e}'} \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{v}^{\mathbf{e}'-\mathbf{e}}.
\end{aligned}$$

Suppose $y(\mathbf{z}) = f(\mathbf{z}, \mathbf{x})$. We have

$$\begin{aligned}
\hat{f}_y(\mathbf{z}, c) &= \sum_{i\in\mathbb{Z}_t} h_i \cdot d! \cdot \left(\sum_{k=1}^{d} \hat{f}_{i,k}(\mathbf{z}, c) - y_i(\mathbf{z}) \right) \\
&= \sum_{i\in\mathbb{Z}_t} h_i \cdot d! \cdot \left(\sum_{k=1}^{d} \sum_{\mathbf{e}\in\mathcal{E}_k} f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}} + \sum_{k=1}^{d} \sum_{\mathbf{e},\mathbf{e}'\in\mathcal{E}_k : \mathbf{e}\neq\mathbf{e}'} \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{v}^{\mathbf{e}'-\mathbf{e}} - y_i(\mathbf{z}) \right) \\
&= \sum_{i\in\mathbb{Z}_t} \sum_{k=1}^{d} \sum_{\mathbf{e},\mathbf{e}'\in\mathcal{E}_k : \mathbf{e}\neq\mathbf{e}'} h_i \cdot d! \cdot \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{v}^{\mathbf{e}'-\mathbf{e}}.
\end{aligned}$$

Let $(\mathbf{u}_0, \mathbf{u}_1) \in \mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$. We have

$$\mathbf{u}_0 = \sum_{i\in\mathbb{Z}_t} \sum_{k=1}^{d} \sum_{\mathbf{e}\neq\mathbf{e}'\in\mathcal{E}_k} h_i \cdot d! \cdot \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{u}_{X^{\mathbf{e}'-\mathbf{e}}} \text{ and}$$
$$\mathbf{u}_1 = \sum_{X_i\in\mathcal{G}_1} x_i \cdot \mathbf{u}_{1,X_i}.$$

We check that the following indeed hold:

$$\begin{aligned}
\mathbf{A}_0 \cdot \mathbf{u}_0 &= \mathbf{A}_0 \cdot \left(\sum_{i\in\mathbb{Z}_t} \sum_{k=1}^{d} \sum_{\mathbf{e}\neq\mathbf{e}'\in\mathcal{E}_k} h_i \cdot d! \cdot \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{u}_{X^{\mathbf{e}'-\mathbf{e}}} \right) \\
&\equiv \sum_{i\in\mathbb{Z}_t} \sum_{k=1}^{d} \sum_{\mathbf{e}\neq\mathbf{e}'\in\mathcal{E}_k} h_i \cdot d! \cdot \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot f_{i,\mathbf{e}}(\mathbf{z}) \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{v}^{\mathbf{e}'-\mathbf{e}} \cdot \mathbf{t}_0 \bmod q \\
&\equiv \hat{f}(\mathbf{z}, c) \cdot \mathbf{t}_0 \bmod q, \\
\mathbf{A}_1 \cdot \mathbf{u}_1 &= \mathbf{A}_1 \cdot \left(\sum_{X_i\in\mathcal{G}_1} x_i \cdot \mathbf{u}_{1,X_i} \right) \equiv \sum_{X_i\in\mathcal{G}_1} x_i \cdot v_i \cdot \mathbf{t}_1 \bmod q \equiv c \cdot \mathbf{t}_1 \bmod q.
\end{aligned}$$

We next analyse the norm of $\mathbf{u}_0$ and $\mathbf{u}_1$. Examining the form $\mathbf{u}_0$ and writing down an upper bound of the norm of each term, we have

$$\mathbf{u}_0 = \underbrace{\sum_{i\in\mathbb{Z}_t}}_{t} \underbrace{\sum_{k=1}^{d}}_{d} \underbrace{\sum_{\mathbf{e}\neq\mathbf{e}'\in\mathcal{E}_k}}_{\binom{w+d}{d}^2} \underbrace{h_i}_{p/2} \underbrace{d! \cdot \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}}}_{(d!)^2} \underbrace{f_{i,\mathbf{e}}(\mathbf{z})}_{(d+1)\cdot\binom{s+d}{d}\cdot\gamma_{\mathcal{R}}^{d}\cdot\alpha^{d+1}} \underbrace{\mathbf{x}^{\mathbf{e}'}}_{\gamma_{\mathcal{R}}^{d-1}\cdot\alpha^{d}} \underbrace{\mathbf{u}_{X^{\mathbf{e}'-\mathbf{e}}}}_{\beta} \cdot$$

Using $\binom{w+d}{d}^2 \leq \frac{(w+d)^{2d}}{(d!)^2}$, $\binom{s+d}{d} \leq \frac{(s+d)^d}{d!}$, and $\frac{d+1}{(d-1)!} \leq 3$, and taking into account the expansion factor $\gamma_{\mathcal{R}}^3$ for multiplying 4 $\mathcal{R}$ elements, we have

$$\|\mathbf{u}_0\| \leq 2 \cdot p \cdot t \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{2d+1} \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2} \leq \delta_0.$$

Similarly, examining $\mathbf{u}_1$, we have

$$\mathbf{u}_1 = \underbrace{\sum_{X_i\in\mathcal{G}_1}}_{w} \underbrace{x_i}_{\alpha} \underbrace{\mathbf{u}_{1,X_i}}_{\beta} \cdot$$

Accounting for the expansion factor $\gamma_{\mathcal{R}}$ for multiplying 2 $\mathcal{R}$ elements, we have $\|\mathbf{u}_1\| \leq w \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}} \leq \delta_1$. □

Theorem 3.5.2. *Let $\mathcal{X}^* := \{ x \in \mathcal{R} : \|x\| \leq \alpha^* \}$. Our VC construction for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ is $\mathcal{X}^*$-extractable if*

$$\begin{aligned}
&\ell_i \geq \mathsf{lhl}(\mathcal{R}, \eta_i, q, \beta) \text{ for } i \in \{0,1\}, \\
&\alpha^* \geq \beta_1^* \geq \delta_1, \\
&\beta_0^* \geq 2 \cdot \max\{ \delta_0, 2 \cdot p \cdot t \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{d+1} \cdot (\alpha^*)^d \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2} \}, \\
&\beta_p^* \geq 2 \cdot \max\{ \delta_p, (s+w+d)^d \cdot \alpha \cdot (\alpha^*)^d \cdot \gamma_{\mathcal{R}}^d \},
\end{aligned}$$

and the k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta_0,\ell_0,w,\mathcal{G}_0,1,\mathcal{D}_0,\mathcal{T}_0,\beta,\beta_0^*}$ *assumption, the knowledge* k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta_1,\ell_1,w,\mathcal{G}_1,\mathcal{D}_1,\mathcal{T}_1,\alpha^*,\beta,\beta_1^*}$ *assumption, and the* R-$\mathsf{SIS}_{\mathcal{R}_p,t,\beta_p^*}$ *assumption hold, where* $\mathcal{D}_i$ *is such that the distribution*

$$\left\{ (\mathbf{A}_i, \mathbf{t}_i, \{\, \mathbf{u}_{\mathcal{G}_i} \,\}, \mathbf{v}) \middle| \begin{array}{l} \mathbf{A}_i \leftarrow \mathcal{R}_q^{\eta_i \times \ell_i};\ \mathbf{t}_i \leftarrow \mathcal{T}_i;\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w \\ \mathbf{u}_g \leftarrow \mathcal{D}_{0,g,\mathbf{A}_i,\mathbf{t}_i,\mathbf{v}},\ \forall g \in \mathcal{G}_i \end{array} \right\}$$

is statistically close to the distribution

$$\left\{ (\mathbf{A}_i, \mathbf{t}_i, \{\, \mathbf{u}_{\mathcal{G}_i} \,\}, \mathbf{v}) \middle| \begin{array}{l} \mathbf{A}_i \leftarrow \mathcal{R}_q^{\eta_i \times \ell_i};\ \mathbf{t}_i \leftarrow \mathcal{T}_i;\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w \\ \mathbf{u}_g \leftarrow \mathsf{SampD}(1^{\eta_i}, 1^{\ell_i}, \mathcal{R}, \beta) : \mathbf{A}_i \cdot \mathbf{u}_g \equiv g(\mathbf{v}) \cdot \mathbf{t}_i \bmod q,\ \forall g \in \mathcal{G}_i \end{array} \right\}.$$

Proof. Suppose $\mathcal{A}$ is a PPT adversary which, on input honestly generated pp and some randomness, outputs $(f, y, \mathbf{z}, c, \pi)$. We construct an extractor $\mathcal{E}_{\mathcal{A}}$ which, on input pp and the same randomness given to $\mathcal{A}$, outputs $\mathbf{x}$.

For the sake of clarity of exposition, let us denote the public parameters pp of the vector commitment scheme as

$$\mathsf{pp} := \begin{pmatrix} \mathsf{pp}_0(\mathbf{v}) := (\mathbf{A}_0, & \mathbf{t}_0, & (\mathbf{u}_{0,g})_{g \in \mathcal{G}_0}, \mathbf{v}), \\ \mathsf{pp}_1(\mathbf{v}) := (\mathbf{A}_1, & \mathbf{t}_1, & (\mathbf{u}_{1,g})_{g \in \mathcal{G}_1}, \mathbf{v}), \mathbf{h} \end{pmatrix},$$

where $\mathsf{pp}_0(\mathbf{v})$ and $\mathsf{pp}_1(\mathbf{v})$ are correlated in that they share the same $\mathbf{v}$.

We define an algorithm $\mathcal{B}^{\mathcal{A}}[\mathsf{pp}]$ which has oracle access to $\mathcal{A}$ and is parameterised by an instance of the VC public parameters $\mathsf{pp} = (\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}_1(\mathbf{v}), \mathbf{h})$. Our algorithm $\mathcal{B}^{\mathcal{A}}[\mathsf{pp}]$ takes as input some $\mathsf{pp}_1'(\mathbf{v}') = (\mathbf{A}_1', \mathbf{t}_1', \left(\mathbf{u}_{1,g}'\right)_{g \in \mathcal{G}_1}, \mathbf{v}')$ and some randomness $r_{\mathcal{A}}$. If $\mathbf{v}' \neq \mathbf{v}$, $\mathcal{B}^{\mathcal{A}}[\mathsf{pp}]$ outputs some arbitrary $(c, \mathbf{u}_1)$. Otherwise, $\mathbf{v}' = \mathbf{v}$, and $\mathcal{B}^{\mathcal{A}}[\mathsf{pp}]$ runs $\mathcal{A}$ on $(\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}_1'(\mathbf{v}), \mathbf{h})$ and the given randomness $r_{\mathcal{A}}$, and obtains $(f, y, \mathbf{z}, c, \pi)$. It parses π as $(\mathbf{u}_0, \mathbf{u}_1)$ and outputs $(c, \mathbf{u}_1)$.

Let $\mathcal{E}^{k\text{-}M\text{-}\mathsf{ISIS}}_{\mathcal{B}^{\mathcal{A}}[\mathsf{pp}]}$ be a PPT extractor whose existence is guaranteed by the knowledge k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta_1,\ell_1,w,\mathcal{G}_1,\mathcal{D}_1,\mathcal{T}_1,\alpha^*,\beta,\beta_1^*}$ assumption. We construct our extractor $\mathcal{E}_{\mathcal{A}}$ as follows.

Our extractor $\mathcal{E}_{\mathcal{A}}$ takes as input some public parameters pp and some randomness $r_{\mathcal{A}}$. Parse $\mathsf{pp} = (\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}_1(\mathbf{v}), \mathbf{h})$. It runs $\mathcal{E}^{k\text{-}M\text{-}\mathsf{ISIS}}_{\mathcal{B}^{\mathcal{A}}[\mathsf{pp}]}$ on input $\mathsf{pp}_1(\mathbf{v})$ and the given randomness $r_{\mathcal{A}}$, and obtains from them a vector $\mathbf{x}$. Finally, $\mathcal{E}_{\mathcal{A}}$ outputs $\mathbf{x}$.

We argue that for $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$, if $(f, y, \mathbf{z}, c, \pi) \leftarrow \mathcal{A}(\mathsf{pp}; r_{\mathcal{A}})$ satisfies $\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, c, \pi) = 1$ with probability ρ, then the probability of $\mathcal{E}_{\mathcal{A}}(\mathsf{td}; r)$ not outputting $\mathbf{x}$ with $\|\mathbf{x}\| \leq \alpha^*$ such that $c = \mathsf{Com}(\mathsf{pp}, \mathbf{x})$ (for some aux suppressed from the output) and $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$ is at most $\kappa(\lambda, s, w, t) = \mathsf{negl}$, where the probabilities are taken over the randomness of Setup and that of $r_{\mathcal{A}}$.

Consider the following hybrid experiments for generating $(\mathsf{pp}, (f, y, \mathbf{z}, c, \pi), \mathbf{x})$ on input $(1^\lambda, 1^s, 1^w, 1^t; (r, r_{\mathcal{A}}))$:

Hyb_0: This is the "real" experiment with procedures as described above. Specifically, it runs $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t; r)$, $(f, y, \mathbf{z}, c, \pi) \leftarrow \mathcal{A}(\mathsf{pp}; r_\mathcal{A})$, and $\mathbf{x} \leftarrow \mathcal{E}_\mathcal{A}(\mathsf{pp}; r_\mathcal{A})$, and outputs $(\mathsf{pp}, (f, y, \mathbf{z}, c, \pi), \mathbf{x})$.

Hyb_1: This experiment is the same as Hyb_0 except that the $\mathsf{pp} = (\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}_1(\mathbf{v}), \mathbf{h})$ passed to $\mathcal{A}$ and $\mathcal{E}_\mathcal{A}$ is replaced by $\mathsf{pp}' = (\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}'_1(\mathbf{v}), \mathbf{h})$ where $\mathsf{pp}'_1(\mathbf{v})$ is sampled as in the definition of k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta_1,\ell_1,w,\mathcal{G}_1,\mathcal{D}_1,\mathcal{T}_1,\alpha^*,\beta,\beta_1^*}$.

Hyb_2: This experiment is the same as Hyb_1 except that the $\mathsf{pp}' = (\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}'_1(\mathbf{v}), \mathbf{h})$ passed to $\mathcal{A}$ and $\mathcal{E}_\mathcal{A}$ is replaced by $\mathsf{pp}'' = (\mathsf{pp}'_0(\mathbf{v}), \mathsf{pp}'_1(\mathbf{v}), \mathbf{h})$ where $\mathsf{pp}'_0(\mathbf{v})$ is sampled as in the definition of k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta_0,\ell_0,w,\mathcal{G}_0,1,\mathcal{D}_0,\mathcal{T}_0,\beta,\beta_0^*}$.

By our assumption on $\mathcal{D}_0$, the distributions Hyb_0 and Hyb_1 are statistically close. Similarly, by our assumption on $\mathcal{D}_1$, the distributions Hyb_1 and Hyb_2 are statistically close. Since the distributions Hyb_0, Hyb_1, and Hyb_2 are all statistically close to each other, for any $i, j \in \mathbb{Z}_3$, if the output of Hyb_i satisfies certain properties with some probability, the output of Hyb_j also satisfies the same properties with similar probability.

The following lemma about the outputs of Hyb_1 is immediate by the knowledge k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta_1,\ell_1,w,\mathcal{G}_1,\mathcal{D}_1,\mathcal{T}_1,\alpha^*,\beta,\beta_1^*}$ assumption.

Lemma 3.5.1. *Let* $(\mathsf{pp}, (f, y, \mathbf{z}, c, \pi), \mathbf{x}) \leftarrow \mathsf{Hyb}_1(1^\lambda, 1^s, 1^w, 1^t; (r, r_\mathcal{A}))$. *Parse* $\mathsf{pp} = (\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}_1(\mathbf{v}), \mathbf{h})$. *If the knowledge* k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta_1,\ell_1,w,\mathcal{G}_1,\mathcal{D}_1,\mathcal{T}_1,\alpha^*,\beta,\beta_1^*}$ *assumption holds, then* $c = \langle \mathbf{v}, \mathbf{x} \rangle \bmod q$ *and* $\|\mathbf{x}\| \leq \alpha^*$ *except with negligible probability.*

The next lemma is about the outputs of Hyb_2.

Lemma 3.5.2. *Let* $(\mathsf{pp}, (f, y, \mathbf{z}, c, \pi), \mathbf{x}) \leftarrow \mathsf{Hyb}_2(1^\lambda, 1^s, 1^w, 1^t; (r, r_\mathcal{A}))$. *Parse* $\mathsf{pp} = (\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}_1(\mathbf{v}), \mathbf{h})$. *If all of the following hold:*

- *the* k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta_0,\ell_0,w,\mathcal{G}_0,1,\mathcal{D}_0,\mathcal{T}_0,\beta,\beta_0^*}$ *assumption,*
- *the* R-$\mathsf{SIS}_{\mathcal{R}_p,t,\beta_p^*}$ *assumption,*
- $c = \langle \mathbf{v}, \mathbf{x} \rangle \bmod q$, *and*
- $\|\mathbf{x}\| \leq \alpha^*$,

then $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$ *except with negligible probability.*

Proof. Parse pp to obtain $(\mathbf{A}_0, \mathbf{t}_0, \mathbf{v}, \mathbf{h})$ and parse π as $(\mathbf{u}_0, \mathbf{u}_1)$. We notice that $\mathbf{h}$ is distributed identically as R-$\mathsf{SIS}_{\mathcal{R}_p,t,\beta_p^*}$ instances. By our assumption on $\mathcal{A}$, with non-negligible probability, it holds that

$$\mathbf{A}_0 \cdot \mathbf{u}_0 \equiv \left(\hat{f}_0(\mathbf{z}, c) - d! \cdot \sum_{i \in \mathbb{Z}_t} h_i \cdot y_i(\mathbf{z}) \right) \cdot \mathbf{t}_0 \bmod q,$$

and $\|\mathbf{u}_0\| \leq \delta_0 \leq \beta_0^*/2$.

Suppose towards a contradiction that the event $f(\mathbf{z}, \mathbf{x}) = \mathbf{y}' \neq y(\mathbf{z})$ for some $\mathbf{y}'$ happens with non-negligible probability. Let $(c', \mathsf{aux}) = \mathsf{Com}(\mathsf{pp}, \mathbf{x})$. By assumption, $c' = c$. Let $(\mathbf{u}_0', \mathbf{u}_1') = \mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$. By a similar calculation as in the proof of correctness (Theorem 3.5.1), it holds that

$$\mathbf{A}_0 \cdot \mathbf{u}_0' \equiv \left(\hat{f}_0(\mathbf{z}, c) - d! \cdot \sum_{i \in \mathbb{Z}_t} h_i \cdot y_i' \right) \cdot \mathbf{t}_0 \bmod q.$$

and

$$\|\mathbf{u}_0'\| \leq 2 \cdot p \cdot t \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{d+1} \cdot (\alpha^*)^d \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2} \leq \beta_0^*/2.$$

Let $\tilde{\mathbf{u}}_0 := \mathbf{u}_0 - \mathbf{u}_0'$. We have

$$\mathbf{A}_0 \cdot \tilde{\mathbf{u}}_0 \equiv d! \cdot \sum_{i \in \mathbb{Z}_t} h_i \cdot (y_i' - y_i(\mathbf{z})) \cdot \mathbf{t}_0 \bmod q.$$

and $\|\tilde{\mathbf{u}}_0\| \leq \beta_0^*$. One (or both) of the following two cases must be true: (i) $\sum_{i \in \mathbb{Z}_t} h_i \cdot (y_i' - y_i(\mathbf{z})) \equiv \mathbf{0} \bmod q$ with non-negligible probability, or (ii) $\sum_{i \in \mathbb{Z}_t} h_i \cdot (y_i' - y_i(\mathbf{z})) \not\equiv \mathbf{0} \bmod q$ with non-negligible probability.

Note that

$$\|\mathbf{y}'\| \leq (s+w+d)^d \cdot \alpha \cdot (\alpha^*)^d \cdot \gamma_{\mathcal{R}}^d \leq \beta_p^*/2$$

and $\|y(\mathbf{z})\| \leq \delta_p \leq \beta_p^*/2$ and hence $\|\mathbf{y}' - y(\mathbf{z})\| \leq \beta_p^*$. If Case (i) is true, we can construct a PPT algorithm for the $R\text{-}\mathsf{SIS}_{\mathcal{R}_p, t, \beta_p^*}$ problem which succeeds with non-negligible probability, which contradicts the $R\text{-}\mathsf{SIS}_{\mathcal{R}_p, t, \beta_p^*}$ assumption.

If Case (ii) is true, we can construct a PPT algorithm for the $k\text{-}M\text{-}\mathsf{ISIS}_{\mathcal{R}_q, \eta_0, \ell_0, w, \mathcal{G}_0, 1, \mathcal{D}_0, \mathcal{T}_0, \beta, \beta_0^*}$ problem which succeeds with non-negligible probability, which contradicts the corresponding assumption.

Since none of the two cases could be true, we must have $f(\mathbf{z}, \mathbf{x}) = \mathbf{y}' = y(\mathbf{z})$. □

Combining the two lemmas, we conclude that for $(\mathsf{pp}, (f, y, \mathbf{z}, c, \pi), \mathbf{x})$ generated by Hyb_0, where $\mathsf{pp} = (\mathsf{pp}_0(\mathbf{v}), \mathsf{pp}_1(\mathbf{v}), \mathbf{h})$, it holds that $c = \langle \mathbf{v}, \mathbf{x} \rangle \bmod q$, $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$, and $\|\mathbf{x}\| \leq \delta_1$ except with negligible probability. □

Theorem 3.5.3. *For $n \in \mathsf{poly}(\lambda)$, $q, \delta_0, \delta_1 \in \mathsf{poly}(\lambda, s, w, t)$, and $\ell_0, \ell_1 \in \Theta(\log q) = \mathsf{polylog}(\lambda, s, w, t)$, covering the choices of parameters in Theorems 3.5.1 and 3.5.2, the VC construction in Figure 3.1 is compact.*

Concretely, let $\mathcal{R}$ be a power-of-2 cyclotomic ring so that $\gamma_{\mathcal{R}} = n$. For $s = w = t \geq n$ and for the following choices of parameters,

$$\begin{aligned}
d, \eta_0, \eta_1 &= O(1), \quad \beta \geq \alpha \\
\delta_0 &= 2 \cdot p \cdot t \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{2d+1} \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2}, \\
\delta_0' &= 2 \cdot p \cdot t \cdot (s+d)^d \cdot (w+d)^{2d} \cdot w^d \cdot \alpha^{2d+1} \cdot \beta^{d+1} \cdot \gamma_{\mathcal{R}}^{3d+2}, \\
\delta_1 &= w \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}, \\
\delta_p' &= (s+w+d)^d \cdot \alpha^{d+1} \cdot (w \cdot \beta \cdot \gamma_{\mathcal{R}}^2)^d \\
p &\approx \delta_p' \cdot n \cdot \log n, \quad q \approx \delta_0' \cdot n \cdot \log n, \ \textit{and} \\
\ell_0 = \ell_1 &= \mathsf{lhl}(\mathcal{R}, 1, q, \beta) \approx 2 \log_\beta q,
\end{aligned}$$

a commitment and openings are of size $O(n \log s)$, and $O(n \cdot (\log s + \log \beta)^2 / \log \beta)$, respectively. The minimum is attained at $\beta = \Theta(s)$, where an opening proof is of size $O(n \log s)$.

Proof. For the general case, we observe that a commitment $c \in \mathcal{R}_q$ is of description size $n \log q \in \mathsf{poly}(\lambda, \log s, \log w, \log t)$, and an opening proof $(\mathbf{u}_0, \mathbf{u}_1)$ is of description size $n \cdot (\ell_0 \log \delta_0 + \ell_1 \log \delta_1) \in \mathsf{poly}(\lambda, \log s, \log w, \log t)$.

For the concrete case, for honestly generated proofs, from Theorem 3.5.1, we have

$$\begin{aligned}
p &\approx \delta_p' \cdot n \cdot \log n = (s+w+d)^d \cdot \alpha^{d+1} \cdot (w \cdot \beta \cdot \gamma_{\mathcal{R}}^2)^d \cdot n \cdot \log n \\
&= O(s^{2d} \cdot \alpha^{d+1} \cdot \beta^d \cdot n^{2d+1} \cdot \log n), \\
\delta_0 &= 2 \cdot p \cdot t \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{2d+1} \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2} \\
&= O(s^d \cdot \alpha^{d+1} \cdot n^{d+1} \cdot \log n) \cdot O(s^{3d+1} \cdot \alpha^{2d+1} \cdot \beta \cdot n^{2d+2}) \\
&= O(s^{4d+1} \cdot \alpha^{3d+2} \cdot \beta \cdot n^{3d+3} \cdot \log n), \\
\delta_0' &= 2 \cdot p \cdot t \cdot (s+d)^d \cdot (w+d)^{2d} \cdot w^d \cdot \alpha^{2d+1} \cdot \beta^{d+1} \cdot \gamma_{\mathcal{R}}^{3d+2} \\
&= O(s^{2d} \cdot \alpha^{d+1} \cdot \beta^d \cdot n^{2d+1} \cdot \log n) \cdot O(s^{4d+1} \cdot \alpha^{2d+1} \cdot \beta^{d+1} \cdot n^{3d+2}) \\
&= O(s^{6d+1} \cdot \alpha^{3d+2} \cdot \beta^{2d+1} \cdot n^{5d+3} \cdot \log n), \\
\delta_1 &= w \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}} = O(s \cdot \alpha \cdot \beta \cdot n), \\
q &\approx \delta_0' \cdot n \cdot \log n = O(s^{6d+1} \cdot \alpha^{3d+2} \cdot \beta^{2d+1} \cdot n^{5d+4} \cdot \log^2 n), \\
\log \delta_0, \log \delta_1, \log q &= O(\log s + \log \alpha + \log \beta + \log n) = O(\log s + \log \beta), \\
\ell_0 = \ell_1 &= 2 \log q / \log \beta = O((\log s + \log \beta) / \log \beta), \\
|c| &= n \cdot \log q = O(n \log s), \ \text{and} \\
|\mathbf{u}_i| &= n \cdot \ell_i \cdot \log \delta_i \\
&= n \cdot O((\log s + \log \beta) / \log \beta) \cdot O(\log s + \log \beta) \\
&= O(n \cdot (\log s + \log \beta)^2 / \log \beta).
\end{aligned}$$

□

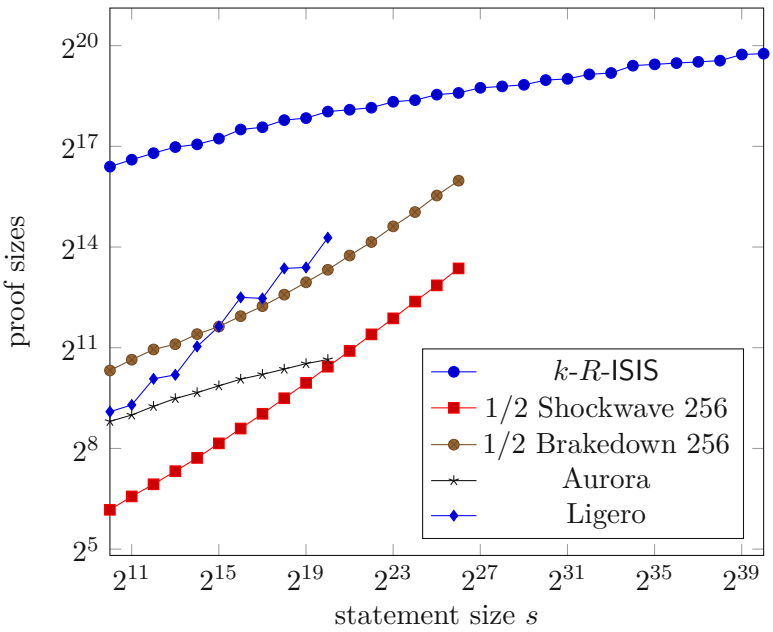

Figure 3.2: Combined size (in KB) of a commitment and an opening proof for the concrete parameters chosen in Theorem 3.5.3, setting $\lambda = 128$, optimising for ρ and comparing with SNARK proof sizes in prior works [GLS+21, Fig. 5]. We picked $\alpha = s$.

Table 3.2: Computation complexities (in number of $\mathcal{R}$ or $\mathcal{R}_q$ operations) of our VC.

Com	$O(w^{2d} \cdot (\log s + \log w + \log t + \log \beta)/\log \beta)$
Open	$O(t \cdot (s+w)^d \cdot (\log s + \log w + \log t + \log \beta)/\log \beta)$
PreVerify	$O(t \cdot (s+w)^d)$
Verify	$O(s^d + (\log s + \log w + \log t + \log \beta)/\log \beta)$

To translate these into concrete sizes we need to pick n such that solving k-R-SIS and R-SIS costs $\approx 2^\lambda$ operations. Here it can be beneficial to set $q = (\delta_0')^\rho \cdot n \cdot \log n$ for some parameter $\rho \in \mathbb{N}$. Specifically, we require that $R\text{-SIS}_{\mathcal{R}_q,\ell_0,2\cdot\sqrt{n}\cdot\delta_0'}$, $R\text{-SIS}_{\mathcal{R}_q,\ell_1,2\cdot\sqrt{n}\cdot\delta_1}$ and $R\text{-SIS}_{\mathcal{R}_p,t,2\cdot\sqrt{n}\cdot\delta_p'}$ are hard. The factor of two arises from our reduction and the factor $\sqrt{n}$ translates between ℓ_∞ and ℓ_2. In Figure 3.2 we report the concrete combined size (in KB) of a commitment and an opening proof for the concrete parameters chosen in Theorem 3.5.3, specifically setting $d = 2$, $\eta_0 = \eta_1 = 1$, and $\beta = s = w = t \in \{2^{10}, 2^{11}, \ldots, 2^{40}\}$.

To analyse computation complexity, we assume the concrete parameter choices in Theorem 3.5.3 with the exception that s, w, t are treated as free variables for more fine-grained complexity measures and to highlight the benefits of preprocessing. For simplicity, we assume $\max\{s, w, t\} \geq n$. The computation complexities (in number of $\mathcal{R}$ or $\mathcal{R}_q$ operations) of Com, Open, PreVerify, and Verify are reported in Table 3.2. Note that

each $\mathcal{R}$ or $\mathcal{R}_q$ operation takes at most $\mathsf{poly}(\lambda, \log s, \log w, \log t)$ time. In summary, the combined time needed to commit to $\mathbf{x}$ and open to $f(\mathbf{z}, \cdot)$ is quasi-quadratic in the time needed to compute $f(\mathbf{z}, \mathbf{x})$, and the time needed to pre-verify (f, y) is quasi-linear in the time needed to compute $f(\mathbf{z}, \mathbf{x})$. We highlight that the online verification cost, i.e. the computation complexity of Verify, is dominated additively by s^d where s is the dimension of the public input. In applications where $s^d = O(\log w + \log t)$ and setting $\beta = \Theta(w + t)$, the online verification cost (in number of bit operations) is $O(n \log w + n \log t)$.

3.6 SNARK for Polynomial Maps Satisfiability

We construct a SNARK Π for $\mathrm{PolySAT}_{\mathcal{R},d,\alpha}$ in Figure 3.3, based on the vector commitment Γ for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ that we developed in Section 3.5. The following theorem establishes the properties of our construction.

$\Pi.\mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$

return $\mathsf{pp} \leftarrow \Gamma.\mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$

$\Pi.\mathsf{PreVerify}(\mathsf{pp}, (f, y))$

return $\mathsf{pp}_{f,y} \leftarrow \Gamma.\mathsf{PreVerify}(\mathsf{pp}, (f, y))$

$\Pi.\mathsf{Prove}(\mathsf{pp}, (f, y, \mathbf{z}), \mathbf{x})$

$(c, \mathsf{aux}) \leftarrow \Gamma.\mathsf{Com}(\mathsf{pp}, \mathbf{x})$
$\pi' \leftarrow \Gamma.\mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$
return $\pi := (c, \pi')$

$\Pi.\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, \pi)$

return $\Gamma.\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, c, \pi')$

Figure 3.3: Construction of SNARK Π for $\mathrm{PolySAT}_{\mathcal{R},d,\beta}$ from a VC Γ for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$.

Theorem 3.6.1. *If Γ is correct, then the SNARK Π presented in Figure 3.3 is complete. If Γ is $\mathcal{X}^*$-extractable, then Π is $\mathcal{X}^*$-knowledge-sound. If the computation complexity of $\Gamma.\mathsf{Verify}$ is in $\mathsf{poly}(\lambda, s, \log w, \log t)$ (implying that Γ is compact), then Π is succinct.*

Proof. (Sketch) Completeness and succinctness are immediate. For knowledge soundness, by the extractability of Γ, for any adversary $\mathcal{A}$ producing a commitment c and a valid opening proof for $(f, y, \mathbf{z})$, there exists an efficient procedure to extract from $\mathcal{A}$ a short vector $\mathbf{x}$ such that $f(\mathbf{z}, \mathbf{x}) = y(\mathbf{z})$, except with negligible probability. □

3.6.1 Proving Relations over $\mathcal{R}_q$

In Section 3.6, we constructed a SNARK for proving knowledge of a short vector $\mathbf{x}$ with $\|\mathbf{x}\| \leq \alpha$ satisfying $f(\mathbf{x}) = \mathbf{y}$, where the polynomial map f and vector $\mathbf{y}$ both have coefficients of norm also at most α.[11] There are, however, natural applications where

[11] We dropped the public input $\mathbf{z}$ for the ease of exposition.

we want to prove algebraic relations which involve $\mathcal{R}$ elements of high norm ($> \alpha$) and where arithmetic is performed modulo q.

For example, the verification equation of a GPV signature [GPV08] is of the form $\mathbf{A} \cdot \mathbf{u} = H(m) \bmod q$, where $\mathbf{A}$ is a random public key matrix over $\mathcal{R}_q$, $H(m)$ is a random vector over $\mathcal{R}_q$ encoding the public message m, and the signature $\mathbf{u}$ is a short vector over $\mathcal{R}$ satisfying the relation. The verification equation of our VC and SNARK constructions $\mathbf{A}_0 \cdot \mathbf{u}_0 \stackrel{?}{=} \hat{f}(c) \cdot \mathbf{t}_0 \bmod q$ have a more complicated form involving the evaluation of a polynomial $\hat{f}$ with large coefficients at a large $\mathcal{R}_q$ element c. In general, consider the task of proving

$$\{ (f, \mathbf{y}) : \exists\ (\mathbf{x}, \mathbf{c}) \in \mathcal{R}^w \times \mathcal{R}_q^\ell,\ f(\mathbf{x}, \mathbf{c}) = \mathbf{y} \bmod q \wedge \|\mathbf{x}\| \leq \delta \}$$

for some $\delta \in \mathbb{R}$ and $q \in \mathbb{N}$, where the polynomial map f and the vector $\mathbf{y}$ have coefficients of norm at most q. Here, $\mathbf{c}$ represent part of the witness which is not necessarily short, e.g. a commitment in our VC construction. We outline a series of transformations on $(f, \mathbf{y})$ and $(\mathbf{x}, \mathbf{c})$ to obtain slightly relaxed[12] statement and witness respectively satisfying a relation natively supported by our SNARK.

We will assume that there exists an odd rational integer $p \in \mathbb{N}$ with $p \leq 2\alpha + 1$ and either $\delta \leq \alpha$ or $2\delta + 1$ is a power of p. Since α for our VC and SNARK and (usually) δ for the application can be chosen freely from a wide range of values, we view this as a mild assumption. The resulting statement and witness will be larger than their original counterparts by a multiplicative factor of $\mathsf{poly}(\log_\alpha q)$.

Handling Modular Reduction. To remove the modular reduction step, let $r \in \mathcal{R}$ be such that $f(\mathbf{x}, \mathbf{c}) + q \cdot r = \mathbf{y}$. Let $q' \in \mathbb{N}$ be the smallest such that $r \in \mathcal{R}_{q'}$. By absorbing r into $\mathbf{c}$ and renaming q' to q, we obtain an equivalent language of the form

$$\{ (f, \mathbf{y}) : \exists\ (\mathbf{x}, \mathbf{c}) \in \mathcal{R}^w \times \mathcal{R}_q^\ell,\ f(\mathbf{x}, \mathbf{c}) = \mathbf{y} \wedge \|\mathbf{x}\| \leq \delta \}\,.$$

Handling Long Witness Components. Next, we transform the witness $(\mathbf{x}, \mathbf{c})$ into an equivalent witness of norm at most α. Write $d = 2\delta + 1$. Let $p \in \mathbb{N}$ be an odd rational integer satisfy the following conditions: (i) $p \leq 2\alpha + 1$, (ii) if $\delta \leq \alpha$ then $p \leq 2\delta + 1$, and (iii) if $\delta > \alpha$ then $p^k = 2\delta + 1 = d$ for some $k \in \mathbb{N}$.

For any $x, h \in \mathbb{N}$, define the p-ary 'Gadget" matrix $\mathbf{G}_{x,h} = (p^i)^{\mathsf{T}}_{i \in \mathbb{Z}_{\lceil \log_p x \rceil}} \otimes \mathbf{I}_h \in \mathcal{R}_q^{h \times h \cdot \lceil \log_p x \rceil}$. Let $\mathbf{G}^{-1}(\cdot)$ denote the component-wise balanced p-ary decomposition, i.e. it outputs a vector with entries in $\{ -(p-1)/2, \ldots, 0, \ldots, (p-1)/2 \}$. Note that $\mathbf{x} = \mathbf{G}_{d,w'} \cdot \mathbf{G}^{-1}(\mathbf{x})$ and $\mathbf{c} = \mathbf{G}_{q,\ell} \cdot \mathbf{G}^{-1}(\mathbf{c})$. By construction, if $\|\mathbf{G}^{-1}(\mathbf{x})\| \leq \alpha$, then we must have $\|\mathbf{x}\| \leq \delta$. Given a polynomial map $f(\mathbf{X}, \mathbf{C})$, define

$$f'(\mathbf{X}', \mathbf{C}') := f(\mathbf{G}_{d,w} \cdot \mathbf{X}, \mathbf{G}_{q,\ell} \cdot \mathbf{C}).$$

[12] In the sense that the norm of the transformed witness has a looser upper bound which is polynomial in the original.

By renaming f' to f and absorbing $\mathbf{c}$ into $\mathbf{x}$, we obtain an equivalent language of the form

$$\{ (f, \mathbf{y}) : \exists\ \mathbf{x} \in \mathcal{R}^w,\ f(\mathbf{x}) = \mathbf{y} \wedge \|\mathbf{x}\| \leq \alpha \}$$

Note that unlike the previous and the original languages f likely contains large coefficients not contained in $\mathcal{R}_q$.

Handling Long Coefficients in Statements. It remains to transform $(f, \mathbf{y})$ with long coefficients to an equivalent statement containing only coefficients of norm at most α. Let $q' \in \mathbb{N}$ be the smallest such that all coefficients of f and $\mathbf{y}$ are contained in $\mathcal{R}_{q'}$. We first replace $f, \mathbf{y}$ by $(f', \mathbf{y}') := (\mathbf{G}^{-1}(f), \mathbf{G}^{-1}(\mathbf{y}))$ where $\mathbf{G}^{-1}(f)$ denotes the coefficient-wise balanced p-ary decomposition of f by viewing f as a linear map on monomials with coefficient vectors in $\mathcal{R}_{q'}^t$. Note that if $\mathbf{x}$ were to satisfy $f'(\mathbf{x}) = \mathbf{y}'$, then it also satisfies $f(\mathbf{x}) = \mathbf{y}$ because

$$\begin{aligned} f'(\mathbf{x}) &= \mathbf{y}' \\ \mathbf{G}^{-1}(f)(\mathbf{x}) &= \mathbf{G}^{-1}(\mathbf{y}) \\ \mathbf{G}_{q',t} \cdot \mathbf{G}^{-1}(f)(\mathbf{x}) &= \mathbf{G}_{q',t} \cdot \mathbf{G}^{-1}(\mathbf{y}) \\ f(\mathbf{x}) &= \mathbf{y}. \end{aligned}$$

However, this transformation is not complete as $f(\mathbf{x}) = \mathbf{y}$ does not necessarily imply $f'(\mathbf{x}) = \mathbf{y}'$.

To address above the issue, we consider any parity-check matrix $\mathbf{H}$ of $\mathbf{G}_{q',t}$, i.e. $\mathbf{G}_{q',t} \cdot \mathbf{H} = \mathbf{0}$ and $\mathbf{H}$ is full-rank. Suppose $\mathbf{x}$ satisfies $f(\mathbf{x}) = \mathbf{y}$. Consider $\mathbf{w} := f'(\mathbf{x}) - \mathbf{y}'$. We have $\mathbf{G} \cdot \mathbf{w} = \mathbf{G} \cdot f'(\mathbf{x}) - \mathbf{G} \cdot \mathbf{y}' = f(\mathbf{x}) - \mathbf{y} = 0$. Therefore there exists unique $\mathbf{z}$ such that $\mathbf{w} = \mathbf{H} \cdot \mathbf{z}$.

With the above observation, we pick a specific $\mathbf{H}$ which has p on the main diagonal, -1 in the entries just below the diagonal and zero everywhere else, and define

$$f''(\mathbf{X}, \mathbf{Z}) := f'(\mathbf{X}) - \mathbf{H} \cdot \mathbf{Z}.$$

By the previous argument, with the knowledge $\mathbf{x}$ satisfying $f(\mathbf{x}) = \mathbf{y}$, one could find a unique $\mathbf{z}$ satisfying $f''(\mathbf{x}, \mathbf{z}) = \mathbf{y}'$. Conversely, suppose $(\mathbf{x}, \mathbf{z})$ satisfies $f''(\mathbf{x}, \mathbf{z}) = \mathbf{y}'$. We have

$$\begin{aligned} f'(\mathbf{x}) - \mathbf{H} \cdot \mathbf{z} &= \mathbf{y}' \\ \mathbf{G}_{q',t} \cdot f'(\mathbf{x}) - \underbrace{\mathbf{G}_{q',t} \cdot \mathbf{H}}_{\mathbf{0}} \cdot \mathbf{z} &= \mathbf{G}_{q',t} \cdot \mathbf{y}' \\ f(\mathbf{x}) &= \mathbf{y}. \end{aligned}$$

Note that the coefficients of f'' and the entries of $\mathbf{y}'$ have norm upper-bounded by α by construction. It remains to upper-bound $\|\mathbf{z}\|$ given that $\|\mathbf{x}\| \leq \alpha$ and $f''(\mathbf{x}, \mathbf{z}) = \mathbf{y}'$.

Let $\mathbf{w} := f'(\mathbf{x}) - \mathbf{y}'$ so that $\mathbf{H} \cdot \mathbf{z} = \mathbf{w}$ and $\|\mathbf{w}\| \leq \alpha' := (w+d)^d \cdot \alpha^{d+1} \cdot \gamma_{\mathcal{R}}^d$. By the construction of $\mathbf{H}$, we have $w_0 = p \cdot z_0$ and $w_i = p \cdot z_i - z_{i-1}$ for $i > 0$. Consequently, we have $\|z_0\| < \|w_0\| \leq \alpha'$ and $\|z_i\| \leq (\|w_i\| + \|z_{i-1}\|)/2 \leq \alpha'$ for $i > 0$.

By renaming f'' to f, $\mathbf{y}'$ to $\mathbf{y}$, and α' to α, and absorbing $\mathbf{z}$ into $\mathbf{x}$, we obtain a relaxed language of the form

$$\{\, (f, \mathbf{y}) : \exists\, \mathbf{x} \in \mathcal{R}^w,\ f(\mathbf{x}) = \mathbf{y} \wedge \|\mathbf{x}\| \leq \alpha \,\}$$

which is natively supported by our SNARK. Note that the resulting language is relaxed in the sense that it only requires $\|\mathbf{x}\|$ to be upper-bounded by $\alpha' = \gamma_{\mathcal{R}}^d \cdot \alpha^{d+1}$ instead of by α required in the original language.

3.6.2 Application

Although a SNARK for an NP-complete language can in principle be used to prove any NP relation, the computation and verification of the proof may not be concretely efficient due to NP reductions. In the following, we highlight a language which is natively supported by our SNARK construction.

Aggregating GPV Signatures.

GPV [GPV08] is a lattice-based signature scheme paradigm of which an instantiation is in the process of being standardised [PFH+20]. GPV signatures are a prime candidate for aggregation as it is unclear how to perform aggregation efficiently in other lattice-signature paradigms based on Schnorr-like paradigms, due to how the random oracle is used there and how it is typically instantiated with hash functions of high multiplicative degree (when viewed as an arithmetic circuit) [DHSS20, BR21, BK20]. On a high level, GPV signatures work as follows. A signature is a short vector $\mathbf{u}$, with respect to a public key $\mathbf{A}$. To verify the signature, the verifier computes $\mathbf{v} = H(m)$, checks that the linear relation $\mathbf{A} \cdot \mathbf{u} \equiv \mathbf{v} \bmod q$ holds and that $\mathbf{u}$ is short, where H is modeled as a random oracle.

As motivated in Section 3.6.1, our SNARK construction can be used to prove knowledge of GPV signatures natively given the verification is a linear relation. The high level idea is to use our SNARK construction to prove knowledge of n signatures where each of them are short vectors satisfying a linear relation. Consider the scenario where the same set of signers, identified with the public keys $(\mathbf{A}_i)_{i \in \mathbb{Z}_n}$, periodically issue signatures $(\mathbf{u}_{i,j})_{i \in \mathbb{Z}_n}$ on a common message m_j with $\mathbf{v}_j = H(m_j)$ at each time j.[13] An aggregator can aggregate the n signatures issued at each time j by computing a SNARK proof for the knowledge of short $(\mathbf{u}_{i,j})_{i \in \mathbb{Z}_n}$ satisfying $\mathbf{A}_i \cdot \mathbf{u}_{i,j} \equiv \mathbf{v}_j \bmod q$. The aggregated signature, i.e. the SNARK proof, can be verified in time sublinear in the number of signers and signatures n by first preprocessing the part of the verification equation depending on

[13] Signing the same message twice produces a solution for M-SIS on $\mathbf{A}_i$, so we may assume a deterministic signature scheme here to avoid this issue.

$(\mathbf{A}_i)_{i\in\mathbb{Z}_n}$. This preprocessing step only needs to be done once for the same set of signers. Then, when the message m_j becomes known at or after time j, the online verification time is only logarithmic in n.

The above idea can also be extended to the case where multiple signers sign different messages. In this case, one can still preprocess the public keys $(\mathbf{A}_i)_{i\in\mathbb{Z}_n}$ of users if they are known ahead of time. The verification time is linear in n since we have to check relations with respect to different messages. However, we are still set to gain from the compactness of the SNARK proof. Such aggregation can aid in the blockchain setting, where an aggregator can aggregate signatures on different transactions included in a block; resulting in smaller blocks to mitigate the effects on the ever-growing size of blockchains.

Recursive SNARK Composition

Since our SNARK construction is purely algebraic over $\mathcal{R}$ and $\mathcal{R}_q$, it can be used to natively prove knowledge of a committed witness and a SNARK proof that satisfy the verification equation. Furthermore, since the verification time of our SNARK construction is sublinear after preprocessing, our SNARK construction can be recursively composed without blowing up the proof size. This makes our SNARK construction suitable for the constructions of verifiable delay functions [BBBF18] and incrementally verifiable computation [Val08] based on the recursive composition of SNARKs. Below, we outline a naive recursive composition strategy which only achieves provable soundness for constant-depth composition. We refer to the literature [Val08, BCCT13, BBBF18] for more advanced composition strategies to support higher-depth composition.

Consider a long computation which involves iteratively applying a computation C on an initial input $\mathbf{x}_0$ for t times to obtain $\mathbf{x}_t = C^t(\mathbf{x}_0)$, where $\mathbf{x}_{i+1} = C(\mathbf{x}_i)$ for $i \in \mathbb{Z}_t$. Let pp be the public parameters sampled by the SNARK construction of a sufficient for the following language $L = L_{\mathsf{pp},C}$ with relation $R = R_{\mathsf{pp},C}$: A statement in L consists of a vector $\mathbf{x}'$. A witness is of the form $(\pi, \mathbf{x})$ where $\mathbf{x}$. The relation R is satisfied if

$$\begin{cases} \pi = \mathbf{x} \ \vee \ \mathsf{Verify}(\mathsf{pp}_R, \mathbf{x}, \pi) = 1 \\ \mathbf{x}' = C(\mathbf{x}) \end{cases}$$

where $\mathsf{pp}_R = \mathsf{PreVerify}(\mathsf{pp}, R)$.

To prove that a statement $(C, \mathbf{x}_t)$ and a witness $\mathbf{x}_0$ satisfy $\mathbf{x}_t = C^t(\mathbf{x}_0)$, the prover computes:

- Set $\mathsf{wit}_0 := \mathbf{x}_0$.
- For $i \in \mathbb{Z}_t$:
  - Compute $\mathbf{x}_{i+1} = C(\mathbf{x}_i)$.
  - Compute $\pi_{i+1} \leftarrow \mathsf{Prove}(\mathsf{pp}, (R, \mathbf{x}_{i+1}), \mathsf{wit}_i)$.
  - Set $\mathsf{wit}_{i+1} := (\pi_{i+1}, \mathbf{x}_{i+1})$

- Output π_t.

The proof can then be verified by checking that $\mathsf{Verify}(\mathsf{pp}_R, \mathbf{x}_t, \pi_t) = 1$.

To show succinctness, we observe that the computation required for checking the relation R given pp_R is of size $\mathsf{polylog}(|R|, |C|, \lambda) \cdot \mathsf{poly}(\lambda) + |C|$, and the computation required for verifying a SNARK proof of the satisfiability of R given pp_R is of size $\mathsf{polylog}(|R|, |C|, \lambda) \cdot \mathsf{poly}(\lambda)$. However, this composition strategy is not known to be provably sound for large t, say $t = \Omega(\lambda)$, since the knowledge extractor may run in time exponential in t (unless the underlying SNARK has a very efficient extractor $\mathcal{E}_{\mathcal{A}}$ which runs only an additive factor longer than the the runtime of $\mathcal{A}$). Fortunately, this issue is discussed and circumvented in many prior works (e.g. [Val08, BCCT13, BBBF18]) where the techniques should also be applicable to the recursive composition of our SNARK construction.

CHAPTER 4

Lattice-Based Succinct Arguments from Vanishing Polynomials

Abstract

Succinct arguments allow a prover to convince a verifier of the validity of any statement in a language, with minimal communication and verifier's work. Among other approaches, lattice-based protocols offer solid theoretical foundations, post-quantum security, and a rich algebraic structure. In this work, we present some new approaches to constructing efficient lattice-based succinct arguments. Our main technical ingredient is a new commitment scheme based on *vanishing polynomials*, a notion borrowed from algebraic geometry. We analyse the security of such a commitment scheme, and show how to take advantage of the additional algebraic structure to build new lattice-based succinct arguments. A few highlights amongst our results are:

1. The first recursive folding (i.e. Bulletproofs-like) protocol for linear relations with *polylogarithmic* verifier runtime. Traditionally, the verifier runtime has been the efficiency bottleneck for such protocols (regardless of the underlying assumptions).
2. The first verifiable delay function (VDF) based on lattices, building on a recently introduced sequential relation.
3. The first lattice-based *linear-time prover* succinct argument for NP, in the pre-processing model. The soundness of the scheme is based on (knowledge)-k-R-ISIS assumption [Albrecht et al., CRYPTO'22].

This chapter presents the result of a collaboration with Russell W. F. Lai and Giulio Malavolta and will be published at the 43rd Annual International Cryptology Conference (CRYPTO'23) under the title "Lattice-Based Succinct Arguments from Vanishing

Polynomials". I am mainly responsible for the design, security proofs, and analysis of the knowledge-based protocols. Further, I contributed to the analysis of the protocol for r1cs. I am also responsible for writing the corresponding sections of the chapter. The accompanying appendix contains extended versions of the constructions and proofs.

4.1 Introduction

A succinct non-interactive argument of knowledge (SNARK) [Kil92, Mic94] allows a prover to convince a verifier that of the validity of an NP relation. The argument is said to be *succinct* if the size of the proof and the runtime of the verifier are sublinear (or, ideally, independent) of the time needed to check the validity of the witness. Due to these strong efficiency requirements, SNARKs for NP have become a cornerstone of modern cryptography: They count a large array of applications [BCG+14, GM17, KMS+16, BGH19, BDFG21b, BMRS20] and have recently found their way into real-world systems in the context of blockchain-based cryptocurrencies [BCG+14, GM17, KMS+16, BGH19, BDFG21b, BMRS20].

A promising approach for constructing efficient SNARKs is to leverage the algebraic structure offered by computational problems in lattice-based cryptography [BISW17, BISW18, GMNO18, BLNS20, AL21, ACK21, ACL+22]. Compared to other approaches (see Section 4.1.2 for a detailed discussion), lattice-based SNARKs offer many desirable properties: (i) They are conjectured to be secure against quantum attacks, (ii) are based on computational problems with solid theoretical foundations, and (iii) have a rich algebraic structure, allowing to prove many interesting statements "natively", i.e. without needing to run the relation through an expensive Karp reduction.

In spite of these promising properties, lattice-based SNARKs are still somewhat limited compared to competing approaches. In particular, known lattice-based schemes suffer from (at least) one of the following limitations:

- They require the verifier to hold a secret information that should not be made available to the prover, i.e. they are in the designated-verifier settings [BISW17, BISW18, GMNO18].
- They have a non-succinct verifier, whose runtime is at least linear in the size of the relation [BLNS20, AL21, ACK21].
- They have a slow prover runtime, i.e. quartic [ACL+22] in the size of the relation.

In this work, we propose new techniques for lattice-based SNARKs that allow us to overcome these barriers, making lattice-based SNARKs qualitatively closer (and, in some aspects, superior) to other approaches.

4.1.1 Our Results

We present new algebraic techniques that allow us to overcome traditional limitations of lattice-based SNARKs. Our central technical ingredient is a new lattice-based commitment scheme based on *vanishing polynomials*, an object borrowed from algebraic geometry. The security of our commitment is based on the vanishing Short Integer Solution (vSIS) problem, a variant of the well-known SIS problem that we introduce in this work. We then show how to exploit the additional algebraic structure of vSIS to obtain new results for lattice-based succinct arguments. In more details, our contributions can be summarized as follows.

(1) The Vanishing-SIS Problem. We introduce the vSIS problem, a variant of the standard SIS over rings, which asks to find a polynomial with short coefficients which vanishes at the given point(s). We show that vSIS is no easier than the k-R-ISIS problem, a recently introduced family of problems [ACL+22]. We also show that vSIS can be explained as a natural generalisation of the search NTRU problem. We propose a worst-case to average-case reduction and a reduction from search NTRU, both conditioning on the hardness of decision NTRU.

(2) New Commitments Based on vSIS. We how the vSIS problem immediately implies the existence of a commitment scheme with useful algebraic properties which are key to our new results in succinct arguments:

- Succinct: The size of the commitment key and the commitment are logarithmic in the size of the input. In particular, this implies that the commitment is also a collision-resistant hash function with very short key.
- Homomorphic: The commitment is (bounded) linearly homomorphic and multiplicatively homomorphic for a constant number of multiplications.
- Foldable: We show that the commitment can be "folded" (in the sense of folding arguments, e.g. Bulletproofs [BLNS20]) in such a way that the folded commitment key retains a succinct representation. Loosely speaking, this allows us to combine the two halves of the committed value and simultaneously half the size of the input *and* the size of the commitment key.

(3) Simple Method for Proving Quadratic Relations. Exploiting the multiplicatively homomorphic property of vSIS commitments, we show a simple method for reducing the task of proving quadratic relations to that of proving linear relations, with only additive quasi-linear overhead in prover time. As an example, to prove that $\langle \mathbf{x}_0, \mathbf{x}_1 \rangle = y$, the prover commits to the polynomials $\bar{p}_{\mathbf{x}_0}(V) = \sum_i x_{0,i} \cdot V^{-i}$ and $p_{\mathbf{x}_1}(V) = \sum_j x_{1,j} \cdot V^j$ as $\bar{c}_{\mathbf{x}_0}$ and $c_{\mathbf{x}_1}$ respectively, and proves the linear relations that the commitments are well-formed. Then, the prover proves that the product $\bar{c}_{\mathbf{x}_0} \cdot c_{\mathbf{x}_1}$, which the verifier can compute themself, is a commitment to a polynomial whose constant term is y, which is

again a linear relation. Instantiating with succinct arguments for linear relations with quasi-linear-time prover, we obtain succinct arguments for quadratic relations also with quasi-linear-time prover.

(4a) New Folding Protocols for Structured SIS. The first kind of linear relations that we consider are *structured SIS relations* (roughly) of the form

$$\begin{pmatrix} \mathbf{A} & & & \\ \mathbf{B} & \mathbf{A} & & \\ & \mathbf{B} & \ddots & \\ & & \ddots & \mathbf{A} \\ & & & \mathbf{B} \\ \mathbf{C}_1 & \mathbf{C}_2 & \ldots\ldots & \mathbf{C}_n \end{pmatrix} \cdot \mathbf{x} = \mathbf{y} \bmod q \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0,$$

where $\mathbf{C}_1, \ldots, \mathbf{C}_n$ conform to certain foldable structure. For such relations, we obtain SNARKs with transparent setup, quasi-linear time prover, and polylogarithmic time verifier (*without* preprocessing), in the random oracle model.[1] The main technical ingredient that enables this result is a new Bulletproof-like folding protocol for *foldable* linear relations, where the verifier runtime is *polylogarithmic* in the length of the relation. Prior folding protocols had a *linear-time verifier* [BLNS20, AL21, ACK21], including those based on the discrete logarithm problem [BCC+16, BBB+18], with the exception of [BMM+21] where the verifier computation is proportional to the square root of the length of the relation.

(4b) Optimised Knowledge-based Protocols for SIS. Next, we consider *unstructured SIS relations* of the form "$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$". For these relations, we obtain SNARKs with quasi-linear time prover and polylogarithmic time verifier after preprocessing, based on the recently introduced (knowledge-)k-R-ISIS assumption [ACL+22]. This improves upon previous schemes which do not natively support proving modular arithmetic relations [ACL+22] and require at least a quadratic-time prover [ACL+22, BCFL22].

(5) Applications. Putting everything together, we obtain SNARKs for quadratic relations with quasi-linear-time prover and polylogarithmic-time verifier (after preprocessing for the unstructured case). We highlight two particular instances.

First, we obtain SNARKs for proving "$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\mathbf{x}$ is *exactly binary*". In particular, applying the structured instantiation on the recently introduced SIS-based sequential relations [LM23], we obtain the first lattice-based verifiable delay functions (VDF). Prior lattice-based schemes [YAZ+19, BLS19, ENS20, LNP22] for exact SIS relations[2] are not succinct.

[1] The interactive variant can be proven secure without random oracles.
[2] not counting those for more general relations

Second, we obtain SNARKs for rank-1 constraint satisfiability (R1CS). Prior lattice-based schemes [ACL+22, BCFL22] have at least quadratic-time prover.

4.1.2 Related Work

There is a vast amount of literature on SNARKs for different classes of relations. We do not attempt to survey all existing works here, but rather provide a high-level overview of various approaches and discuss in details those that are closely related to our work.

Pairing-based. To date, the most efficient and feature-rich SNARKs are constructed over *bilinear pairing groups* (e.g. [Gro16]) with a trusted setup. Typically, they are publicly verifiable and have simple verification algorithms consisting of a constant amount of pairing-product equations. Moreover, pairing-based SNARKs offer a rich algebraic structures that is known to enable proof batching [LMR19, BMM+21] and efficient recursive composition [BCTV14a].

Hash-based. Another approach to build SNARKs is to compile an information-theoretic proof system, e.g. a probabilistically checkable proof (PCP) [Kil92, Mic94] or an interactive oracle proof (IOP), via a vector commitment scheme. Since the vector commitment is usually instantiated with a Merkle-hash tree in the random oracle (RO) model, we call this the hash-based approach. A major difference between pairing-based and hash-based SNARKs, from both theoretical and practical perspectives, is the algebraic structure of the verification algorithm. The reliance of hash-based SNARKs on an RO makes recursive composition challenging, since an RO is typically instantiated with a hash function of high multiplicative degree. On the flip side, hash-based SNARKs can be shown to be post-quantum secure [CMS19].

Lattice-based. Finally, we discuss *lattice-based* approaches to build SNARKs. Until recently, lattice-based SNARKs required the verifier to keep a secret state hidden from the prover, i.e. they are in the designated verifier settings [GMNO18, ISW21]. Excitingly, recent development sees two emerging paradigms for constructing publicly verifiable SNARKs, both of which we improve upon in this work.

The first line of work [BLNS20, AL21, ACK21] studies lattice-based folding protocols which, as discussed above, give quasi-linear-time prover SNARKs in the random oracle model. However, due to lack of preprocessing support, the verifier complexity in folding protocols has always been is *linear* in the size of the relation. In this work, we work around this barrier by considering structured relations which retain their foldable structures after folding, and obtain the first folding protocols with a polylogarithmic-time verifier.

Another line of work [ACL+22, BCFL22] constructs publicly verifiable SNARKs in the preprocessing model. At the core of these constructions are functional commitment schemes which allow to succinctly prove that a committed vector $\mathbf{x}$ satisfies $f(\mathbf{x}) = \mathbf{y}$ for low-degree polynomials [ACL+22] or even unbounded-depth circuits [BCFL22]. To this

end, we propose a construction with *quasi-linear*-time prover using similar techniques, while in [ACL+22, BCFL22] the prover has at least quadratic complexity. We remark that while the recent work of Wee and Wu [WW23b] constructs functional commitments for circuits, their scheme does not support preprocessing and therefore has inefficient verifier.

4.2 Technical Overview

We provide a high-level overview of the techniques that we develop in this work. First, we present our main new technical ingredient that is at the centre of our results, namely a new commitment based on vanishing-SIS. Then we show how arguments for vanishing-SIS commitments can be efficiently composed into an argument for binary-satisfiability of both structured and unstructured linear relations. Finally, we describe our new succinct arguments in both the structured and unstructured settings, and present some immediate applications.

Throughout this overview, we will work with a cyclotomic field $\mathcal{K} = \mathbb{Q}(\zeta)$ where ζ is a root of unity of some prime order ρ, its ring of integers $\mathcal{R} = \mathbb{Z}[\zeta]$, and the quotient rings $\mathcal{R}_q \coloneqq \mathcal{R}/q\mathcal{R}$ for different values of $q \in \mathbb{N}$. Ring elements will be represented by their coefficient embedding and the norm of a ring element is defined accordingly. Readers not familiar with these objects can treat $\mathcal{K} = \mathbb{Q}$ and $\mathcal{R} = \mathbb{Z}$, which suffices in most places.

4.2.1 Vanishing-SIS Commitments

The main technical ingredient behind of our results is a new family of commitment schemes for committing to short vectors $\mathbf{x} \in \mathcal{R}^d$ and companion argument systems for proving that the committed vector is in fact a bit string, i.e. $\mathbf{x} \in \{0,1\}^d$. In their simplest form, the commitment key is a single random element $v \leftarrow \mathcal{R}_q^\times$, where $\mathcal{R}_q^\times$ is the set of invertible elements in $\mathcal{R}_q$. To commit to a *short* $\mathbf{x} \in \mathcal{R}^d$, we interpret $\mathbf{x}$ as the coefficients of a degree-d homogeneous polynomial $p_{\mathbf{x}}(V)$ and compute the commitment as the evaluation of $p_{\mathbf{x}}$ at the point v modulo q, i.e.

$$p_{\mathbf{x}}(V) = \sum_{i=1}^{d} x_i \cdot V^i \qquad \text{and} \qquad c = p_{\mathbf{x}}(v) \bmod q.$$

We refer to this family of commitment schemes as the vanishing short integer solution (vSIS) commitments, for reasons that will become clear shortly. The binding property of the vSIS commitment above is based on the following vSIS assumption which we introduce in this work.

Definition 4.2.1 (vSIS, Informal)**.** *Given a random point $v \leftarrow \mathcal{R}_q^\times$, it is hard to find a degree-d polynomial $p = \sum_{i=0}^{d} p_i \cdot V^i \in \mathcal{R}[V]$ with short coefficients such that $p(v) = 0 \bmod q$. In other words, p is a short element in $\mathcal{I}(v)$, the ideal (lattice) of polynomials vanishing at the given point v.*

In general, the vSIS assumption could be parametrised by a set $\mathcal{G}$ of (multivariate) monomials[3] over $\mathcal{R}$, where the task is to find a short linear combination $(p_g)_{g\in\mathcal{G}}$ such that $\sum_{g\in\mathcal{G}} p_g \cdot g(\mathbf{v}) = 0 \bmod q$. To gain confidence in its validity, we show that the vSIS assumptions are implied by the k-R-ISIS assumptions recently introduced in [ACL+22]. For certain parameter regimes (although *not* the ones that we consider in this work), we show that the vSIS problem is as hard as the search NTRU problem, conditioned on the hardness of the decision NTRU problem. For more details, we refer the reader to Section 4.4 and Section 4.5.

The vSIS commitment schemes have nice homomorphic properties. For starters, they are clearly linearly homomorphic, similarly to the standard SIS-based commitments. More importantly for us, they are also bounded *multiplicatively homomorphic*: If c_f and c_g commit to the polynomials f and g respectively, then $c_f \cdot c_g \bmod q$ commits to the polynomial $f \cdot g$. An elementary fact that will be particularly useful later, is that if $g(V) = f(V^{-1})$, then the constant term of $f \cdot g$ is given by the inner-product of the coefficients of f and g.

Proof of Binary-Satisfiability of Linear Relations. As a warm-up, we outline the construction of a succinct argument system for a prover to convince a verifier that a vector $\mathbf{x} \in \mathcal{R}^d$ satisfies

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \qquad \text{and} \qquad \mathbf{x} \in \{0,1\}[d].$$

As building blocks, we will use succinct argument systems for SIS relations with soundness gaps, i.e., they are complete and sound for relations of the form

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0,$$

but the constraints on the shortness of $\mathbf{x}$ differ. That is, we will turn succinct arguments for showing that $\mathbf{x}$ satisfying a linear relation is short, into an argument for showing the $\mathbf{x}$ is *exactly binary*. While this may seem like a technicality, this *proof of binariness* will be crucial for our later applications, and can be generalised to proof arbitrary quadratic relations. Later in this overview, we will also show how to instantiate the required building blocks.

The common reference string of our argument system contains a random vector $\mathbf{h} \in \mathcal{R}_{q_1}^d$ and a vSIS commitment key $v \in \mathcal{R}_{q_2}^\times$, where $q_0 \ll q_1 \ll q_2$ and the purpose of $\mathbf{h}$ will become clear later. For $\mathbf{x} \in \mathcal{R}^d$ and $\mathbf{w} = (\mathbf{w}_-, \mathbf{w}_+) \in \mathcal{R}^{2d}$, define the (Laurent) polynomials

$$\bar{p}_{\mathbf{x}}(V) \coloneqq p_{\mathbf{h}\circ\mathbf{x}}(V^{-1}) \qquad \text{and} \qquad \tilde{p}_{\mathbf{w}}(V) \coloneqq p_{\mathbf{w}_-}(V^{-1}) + p_{\mathbf{w}_+}(V),$$

where $\mathbf{h} \circ \mathbf{x}$ denotes the Hadamard (component-wise) product of the two vectors. The argument proceeds as follows:

[3] Or rational functions in general.

1. The prover reveals the following "complementary" vSIS commitments to $\mathbf{x}$:

$$c_{\mathbf{x}} \coloneqq p_{\mathbf{x}}(v) \bmod q_2 \qquad \text{and} \qquad \bar{c}_{\mathbf{x}} \coloneqq \bar{p}_{\mathbf{x}}(v) \bmod q_2.$$

2. The prover then proves the following relations:

$$\exists\ \mathbf{x} \in \mathcal{R}^d, \quad \begin{aligned} \mathbf{M} \cdot \mathbf{x} &= \mathbf{y} \bmod q_0, \\ p_{\mathbf{x}}(v) &= c_{\mathbf{x}} \bmod q_2, \\ \bar{p}_{\mathbf{x}}(v) &= \bar{c}_{\mathbf{x}} \bmod q_2, \end{aligned} \quad \text{and} \quad \|\mathbf{x}\| \approx 0. \tag{4.1}$$

$$\exists\ \mathbf{w} \in \mathcal{R}^{2d}, \quad \tilde{p}_{\mathbf{w}}(v) = c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - \bar{p}_{\mathbf{1}}(v)) \bmod q_2 \quad \text{and} \quad \|\mathbf{w}\| \approx 0. \tag{4.2}$$

Since $p_{\mathbf{x}}(v)$, $\bar{p}_{\mathbf{x}}(v)$, and $\tilde{p}_{\mathbf{w}}(v)$ can be computed as linear functions evaluated at the monomials expansion of v, Equations (4.2) and (4.6) can be proven by using argument systems for SIS relations, as required above.

The interesting bit of our protocols is that, even though the the underlying arguments for the SIS relation have soundness gaps, the verifier of our protocol will be convinced that $\mathbf{x}$ is *exactly* binary. First, from the knowledge soundness of the argument for Equation (4.6), the verifier is convinced that there exists a candidate short vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{w}}$ satisfying Equation (4.6) and Equation (4.2) respectively. From $\hat{\mathbf{x}}$, one could derive a short vector $\hat{\mathbf{u}} = (\hat{\mathbf{u}}_-, \hat{u}_0, \hat{\mathbf{u}}_+) \in \mathcal{R}^{2d+1}$ encoding

$$\hat{p}_{\hat{\mathbf{u}}}(V) \coloneqq p_{\mathbf{x}}(V) \cdot \bar{p}_{\mathbf{x}-\mathbf{1}}(V) = p_{\mathbf{x}}(V) \cdot p_{\mathbf{h}\circ(\mathbf{x}-\mathbf{1})}(V^{-1}).$$

Clearly, $\hat{p}_{\hat{\mathbf{u}}}(v) = c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - \bar{p}_{\mathbf{1}}(v)) \bmod q_2$. This means that $\tilde{p}_{\hat{\mathbf{w}}}(V) - \hat{p}_{\hat{\mathbf{u}}}(V)$ is a polynomial with short coefficients which vanishes at v. Furthermore, notice that $\tilde{p}_{\hat{\mathbf{w}}}$ does not have a constant term, while the constant term $\hat{u}_0$ of $\hat{p}_{\hat{\mathbf{u}}}$ is given by the inner-product

$$\hat{u}_0 = \langle \mathbf{x}, \mathbf{h} \circ (\mathbf{x} - \mathbf{1}) \rangle = \sum_{i=1}^{d} h_i \cdot \underbrace{x_i \cdot (x_i - 1)}_{=0 \text{ iff } x_i \in \{0,1\}}.$$

Let us first establish that $\hat{u}_0$ must indeed be 0. This is an easy reduction to the vSIS, since it would otherwise yield a non-zero short solution to a vSIS problem, which we assume to be hard to find. However, we are not yet done, since the fact that $\hat{u}_0 = 0$ *does not* imply that all of its summands are also zero (which is what we need to ensure that $\mathbf{x}$ is binary). This is where the vector $\mathbf{h}$ comes into play, using a technique first introduced in [ACL+22]: Suppose $\hat{u}_0 = 0$, then we also have $\hat{u}_0 = \sum_{i=1}^{d} h_i \cdot x_i \cdot (x_i - 1) = 0 \bmod q_1$. If $\mathbf{x}$ is not binary, the vector $\mathbf{x} \circ (\mathbf{x} - \mathbf{1})$ would be a short non-zero solution to the RingSIS instance given by $\mathbf{h}$ over $\mathcal{R}_{q_1}$.

4.2.2 Efficient Proofs for SIS Relations

In the above proof of binary-satisfiability of linear relations, the prover and verifier computation costs are dominated by the costs of the succinct arguments for SIS relations with soundness gaps. Here we discuss two approaches in the literature, and how we can improve on both fronts using the algebraic properties of our vSIS-based commitment scheme.

Approach I: Folding Protocols. (Lattice-based) Bulletproofs [BLNS20, AL21, ACK21] are interactive arguments with quasi-linear time prover, and can be made non-interactive using the Fiat-Shamir transform in the random oracle model. It is based on the technique of iteratively "folding" the relation into a smaller one until a trivial relation is derived. Recall that in Bulletproofs the prover wants to convince the verifier that they know a short vector $\mathbf{x}$ satisfying

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0.$$

Let $(\mathbf{M}, \mathbf{x}, \mathbf{y}) = (\mathbf{M}^{(0)}, \mathbf{x}^{(0)}, \mathbf{y}^{(0)})$. The protocol consists of $\ell + 1$ rounds, where in the i-th round the two parties "fold" the relation represented by $(\mathbf{M}^{(i)}, \mathbf{y}^{(i)})$ into another represented by $(\mathbf{M}^{(i+1)}, \mathbf{y}^{(i+1)})$ where the dimension of $\mathbf{M}^{(i+1)}$ is half that of $\mathbf{M}^{(i)}$. Correspondingly, the prover folds its witness $\mathbf{x}^{(i)}$ into $\mathbf{x}^{(i+1)}$. After ℓ such folding steps, a constant-size relation $(\mathbf{M}^{(\ell)}, \mathbf{y}^{(\ell)})$ is reached and the prover simply sends the satisfying witness $\mathbf{x}^{(\ell)}$ over to the verifier.

In more detail, each of the first ℓ rounds of the protocol goes as follows. For the i-th round, $i \in \{0, \ldots, \ell - 1\}$, the parties split $\mathbf{M}^{(i)}$ into two halves as $\mathbf{M}^{(i)} = (\mathbf{M}_L^{(i)}, \mathbf{M}_R^{(i)})$ and the prover further splits $\mathbf{x}^{(i)} = (\mathbf{x}_L^{(i)}, \mathbf{x}_R^{(i)})$. The prover sends the cross terms

$$\mathbf{y}_{LR}^{(i)} = \left\langle \mathbf{M}_L^{(i)}, \mathbf{x}_R^{(i)} \right\rangle \bmod q \qquad \text{and} \qquad \mathbf{y}_{RL}^{(i)} = \left\langle \mathbf{M}_R^{(i)}, \mathbf{x}_L^{(i)} \right\rangle \bmod q.$$

The verifier sends a random challenge $r_i \leftarrow S$ sampled from some challenge set $S \subseteq \mathcal{R}^\times$. Both parties fold $(\mathbf{M}^{(i)}, \mathbf{y}^{(i)})$ into

$$(\mathbf{M}^{(i+1)}, \mathbf{y}^{(i+1)}) \coloneqq (\mathbf{M}_L^{(i)} + \mathbf{M}_R^{(i)} \cdot r_i^{-1}, \mathbf{y}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{y}^{(i)} + \mathbf{y}_{LR}^{(i)} \cdot r_i) \bmod q,$$

and the prover folds $\mathbf{x}$ into $\mathbf{x}^{(i+1)} = \mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i$. At the ℓ-th (i.e. last) round, the prover simply sends $\mathbf{x}^{(\ell)}$ and the verifier checks that $\mathbf{x}^{(\ell)}$ is short and satisfies $\left\langle \mathbf{M}^{(\ell)}, \mathbf{x}^{(\ell)} \right\rangle = \mathbf{y}^{(\ell)} \bmod q$.

It can been shown [BLNS20, AL21, ACK21, AF22] that the protocol satisfies knowledge soundness, and furthermore it is easy to see that the prover runs in time quasi-linear in the length of the witness. However, a major drawback of this approach is that the verifier computation is also quasi-linear for general linear relations $\mathbf{M}$, and it cannot be preprocessed due to the interactive nature of the scheme.

Polylogarithmic Verifier for Structured Relations. In this work, we observe that, while we cannot hope to reduce the verifier complexity for general matrices $\mathbf{M}$, for suitably structured $\mathbf{M}$ the verification can be sped up to run in time polylogarithmic in the witness length. As an example, the simplest $\mathbf{M}$ with the required structure is a vector consisting of powers of an element $v \in \mathcal{R}_q^\times$, i.e.

$$\mathbf{M} = \begin{pmatrix} v & v^2 & \ldots & v^d \end{pmatrix} \bmod q.$$

Importantly for us, $\langle \mathbf{M}, \mathbf{x} \rangle = p_{\mathbf{x}}(v) \bmod q$ is exactly the vSIS commitment of $\mathbf{x}$ with commitment key v. Thus, this observation allows us to prove the knowledge of a pre-image of a vSIS commitment via the above protocol with polylogarithmic verifier complexity.

To see why this is the case, it suffices to observe that the verifier complexity is dominated by the computation of the matrix $\mathbf{M}^{(\ell)}$, which is obtained by successive foldings of the starting matrix $\mathbf{M}^{(0)}$. Plugging in the structured relation, we can see that at each iteration the matrix evolves into

$$\begin{aligned}\mathbf{M}^{(i+1)} &= \mathbf{M}_L^{(i)} + \mathbf{M}_R^{(i)} \cdot r_i^{-1} \\ &= \begin{pmatrix} v & v^2 & \dots & v^{d_i/2} \end{pmatrix} + \begin{pmatrix} v^{d_i/2+1} & v^{d_i/2+2} & \dots & v^{d_i} \end{pmatrix} \cdot r_i^{-1} \\ &= \begin{pmatrix} v & v^2 & \dots & v^{d_i/2} \end{pmatrix} \cdot (1 + v^{d_i/2} \cdot r_i^{-1}) \bmod q\end{aligned}$$

where d_i is the input length at the i-th iteration. Recursing over all iterations, we obtain that the final matrix $\mathbf{M}^{(\ell)}$ is defined as

$$\mathbf{M}^{(\ell)} = \prod_{i=0}^{\ell-1} \left(1 + v^{2^{\ell-i-1}} \cdot r_i^{-1}\right) \bmod q,$$

which can be computed in time polynomial in ℓ, i.e. polylogarithmic in d. In Sections 4.6 and 4.7, we extend the above structured folding technique in three ways:

1. We identify a general class of "foldable" (block-)matrices for which the verifier computation can be made polylogarithmic in the number of columns.
2. By modifying the Bulletproofs protocol with techniques borrowed from another folding protocol of Pietrzak [Pie19], we are able to support foldable matrices with an arbitrary (i.e non-power-of-2) number of columns, without breaking the foldable structure.[4]
3. Borrowing techniques from [Pie19] again, we can make the verifier computation *also* polylogarithmic in the number of rows of $\mathbf{M}$, for $\mathbf{M}$ with repeating block-bidiagonals, if $\mathbf{y}$ is also foldable.

Approach II: Pre-Processing (Knowledge-Based) Protocols. The second approach for lattice-based arguments for SIS relation is the recent work of [ACL+22], which is based on a newly recently introduced (knowledge-)k-R-ISIS assumption. In this protocol, the verifier computation can be preprocessed such that the online verification time is polylogarithmic in the relation size. However, a major drawback of this approach is that the public parameters size and the prover complexity are at least quadratic in the relation size. Let us recall (a somewhat simplified version of) the commit-and-prove

[4] The usual technique of padding zero columns breaks the foldable structure.

protocol of [ACL+22] specialised to the case of SIS (i.e. linear) relations. The public parameters consists of

$$\mathbf{A}, \mathbf{t}, \mathbf{v}, \mathbf{h}, \left(\mathbf{A}^{-1}(\mathbf{t} \cdot (g \cdot \bar{g}')(\mathbf{v}))\right)_{g,g' \in \mathcal{G}, g \neq g'}$$

for some set of monomials $\mathcal{G}$, where $\mathbf{A}, \mathbf{t}, \mathbf{v}$ are random over $\mathcal{R}_{q_2}$, $\mathbf{h}$ is a random vector over $\mathcal{R}_{q_1}$, $\bar{g} \coloneqq 1/g$ denotes the complement of g, and $\mathbf{A}^{-1}(\mathbf{t} \cdot g(\mathbf{v}))$ denotes a short preimage $\mathbf{u}_g$ satisfying $\mathbf{A} \cdot \mathbf{u}_g = \mathbf{t} \cdot g(\mathbf{v}) \bmod q_2$. To prove that

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \text{ (without mod)} \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0,$$

commit to $\mathbf{x}$ as $c_{\mathbf{x}} \coloneqq \sum_{g \in \mathcal{G}} x_g \cdot g(\mathbf{v}) \bmod q_2$ and derive a short vector $\mathbf{u}$ satisfying

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{t} \cdot \mathbf{h}^{\mathsf{T}} \cdot (\mathbf{M} \cdot \bar{\mathcal{G}}(\mathbf{v}) \cdot c_{\mathbf{x}} - \mathbf{y}) \bmod q_2,$$

where $\bar{\mathcal{G}}(\mathbf{v}) = (\bar{g}(\mathbf{v}))_{g \in \mathcal{G}}$. To compute such a short vector $\mathbf{u}$, the prover needs to perform a linear combination of $|\{ g \cdot \bar{g}' : g, g' \in \mathcal{G}, g \neq g' \}|$ short vectors given in the public parameters. For $\mathcal{G} = \{ V_1, \ldots, V_d \}$ chosen in [ACL+22], we have $|\{ g \cdot \bar{g}' : g \neq g' \in \mathcal{G} \}| = O(d^2)$, hence the quasi-quadratic prover complexity.

Achieving Quasi-Linear Time Prover. A natural idea is to choose $\mathcal{G} = \{ V, V^2, \ldots, V^d \}$ so $\mathbf{v}$ becomes a single element v. This makes

$$|\{ g \cdot \bar{g}' : g, g' \in \mathcal{G}, g \neq g' \}| = |\{ V^{-i}, V^i \}_{i=1}^{d-1}| = 2d - 2 = O(d).$$

Further exploiting fast multiplication algorithms for Toeplitz matrices allows us to achieve quasi-linear prover time. Notably, with this choice of $\mathcal{G}$ we have

$$c_{\mathbf{x}} = p_{\mathbf{x}}(v) \bmod q_2 \qquad \text{and} \qquad \mathbf{h}^{\mathsf{T}} \cdot \mathbf{M} \cdot (\bar{g}(v))_{g \in \mathcal{G}} = \bar{p}_{\mathbf{M}^{\mathsf{T}} \cdot \mathbf{h}}(v) \bmod q_2,$$

and $\mathbf{h}^{\mathsf{T}} \cdot \mathbf{M} \cdot (\bar{g}(V))_{g \in \mathcal{G}} \cdot c_{\mathbf{x}} - \mathbf{h}^{\mathsf{T}} \cdot \mathbf{y}$ being a polynomial with constant term 0. In the main body, we also show how to support natively modular arithmetic, by borrowing techniques from chainable functional commitments [BCFL22]. We refer the interested reader to Section 4.8 for more details.

4.2.3 Applications

To summarise, we have constructed succinct arguments for relations of the form

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \qquad \text{and} \qquad \mathbf{x} \in \{0,1\}^d$$

with quasi-linear time provers (in both the folding and the preprocessing settings). This gives a efficient and powerful building block for constructing advanced lattice-based cryptographic primitives which require proving relations of the above form. We provide a few examples below.

Lattice-based Verifiable Delay Functions. For the instantiation based on folding protocols, the verifier computation is polylogarithmic if the relation $(\mathbf{M}, \mathbf{y})$ conforms to a certain foldable structure. One example is the sequential-SIS relation proposed in a recent work [LM23], which was used to construct proofs of sequential work (PoSW). In more details, the sequential-SIS relation proposed in their work induces the following linear relation

$$\underbrace{\begin{pmatrix} \mathbf{G} & & & \\ \mathbf{A} & \mathbf{G} & & \\ & \mathbf{A} & \ddots & \\ & & \ddots & \mathbf{G} \\ & & & \mathbf{A} \end{pmatrix}}_{\mathbf{M}} \cdot \mathbf{x} = \underbrace{\begin{pmatrix} \mathbf{z}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{z}_T \end{pmatrix}}_{\mathbf{y}} \bmod q \qquad \text{and} \qquad \mathbf{x} \in \mathcal{R}_2^{mT},$$

for a uniformly sampled $\mathbf{A}$ and $\mathbf{z}_0$. The PoSW construction in [LM23] falls short of giving verifiable delay functions (VDF) due to the soundness gap in lattice-based folding protocols. By embedding the $\mathbb{Z}_2$ coefficients of $\mathbf{x} \in \mathcal{R}_2^{mT}$ into $\mathbf{x}' \in \{0,1\}^{mT\varphi(\rho)}$, and plugging in the structured folding protocol constructed in this work, we immediately get the first construction of lattice-based VDFs.

Efficient Lattice-based SNARKs for NP. Recall that our results ultimately rely on the observation that the inner-product of $\mathbf{x}$ and $\mathbf{y}$ is encoded as the constant term of the polynomial $p_{\mathbf{x}} \cdot \bar{p}_{\mathbf{y}}$. In the above, we used this to encode the vectors $\mathbf{x}$ and $\mathbf{y} \coloneqq \mathbf{h} \circ (\mathbf{x} - \mathbf{1})$ for proving binariness. The same idea can be used to prove general quadratic relations.

Consider the NP-complete rank-1 constraint satisfiability (R1CS) relation which is of the form

$$\exists\ \mathbf{x},\ (\mathbf{A} \cdot \mathbf{x}) \circ (\mathbf{B} \cdot \mathbf{x}) = \mathbf{C} \cdot \mathbf{x} \bmod q$$

where some entries of $\mathbf{x}$ are publicly known. To prove knowledge of $\mathbf{x}$, the prover computation roughly goes as follows. First, they compute

$$\mathbf{a} \coloneqq \mathbf{A} \cdot \mathbf{x}, \qquad \mathbf{b} \coloneqq \mathbf{B} \cdot \mathbf{x}, \qquad \text{and} \qquad \mathbf{c} \coloneqq \mathbf{C} \cdot \mathbf{x}.$$

They then commit to $(\mathbf{x}, \mathbf{h} \circ \mathbf{a}, \mathbf{b}, \mathbf{c})$ as $(c_{\mathbf{x}}, \bar{c}_{\mathbf{a}}, c_{\mathbf{b}}, c_{\mathbf{c}})$, and prove that the commitments are consistent. Finally, they prove that the constant term in (the polynomial underlying) $\bar{c}_{\mathbf{a}} \cdot c_{\mathbf{b}}$ is identical to $\langle \mathbf{h}, \mathbf{c} \rangle$ for $\mathbf{c}$ committed in $c_{\mathbf{c}}$.

4.3 Preliminaries

Let $\lambda \in \mathbb{N}$ denote the security parameter, and $\mathsf{poly}(\lambda)$ and negl the set of all polynomials and negligible functions in λ respectively. Denote the empty string by ϵ. For a function f which may depend on λ and other parameters, we write $O_\lambda(f) \coloneqq f \cdot \mathsf{poly}(\lambda)$ to hide fixed polynomial factors in λ. For matrices $\mathbf{A}$ and $\mathbf{B}$ with the same dimensions, write

$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow 3} := \begin{pmatrix}\mathbf{A} & & \\ \mathbf{B} & \mathbf{A} & \\ & \mathbf{B} & \mathbf{A} \\ & & \mathbf{B}\end{pmatrix}$. The notation $\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n}$ is defined analogously for any $n \in \mathbb{N}$. If S is a set and $\mathcal{D}$ is a distribution over S, we write $\mathcal{D} \sim S$.

4.3.1 Cyclotomic Rings

Let $\mathcal{K} = \mathbb{Q}(\zeta)$ be a cyclotomic field, where ζ is a root of unity of order $\rho = \mathsf{poly}(\lambda)$, and $\mathcal{R} = \mathbb{Z}[\zeta]$ be its ring of integers. If ρ is a power of 2 (resp. prime power), $\mathcal{R}$ is called a power-of-2 (resp. prime power) cyclotomic ring. For $q \in \mathbb{N}$, define the quotient ring $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. We denote by $\mathcal{R}^\times$ and $\mathcal{R}_q^\times$ the sets of units in $\mathcal{R}$ and $\mathcal{R}_q$ respectively. An element $a = \sum_{i=0}^{\rho-1} a_i \cdot \zeta^i \in \mathcal{R}$ (or $\mathcal{R}_q$) is represented by its coefficients $(a_0, \ldots, a_{\rho-1}) \in \mathbb{Z}^\rho$ (or $\mathbb{Z}_q^\rho$). The (infinity) norm of $a \in \mathcal{R}$ (or $\mathcal{R}_q$) is taken as $\|a\| := \max_{i=0}^{\rho-1}(|a_i|)$, where in the case of $a_i \in \mathbb{Z}_q$ the balanced representation is taken, i.e. $a_i \in \{-\lceil q/2\rceil + 1, \ldots, \lfloor q/2\rfloor\}$. For a vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathcal{R}^n$, $\|\mathbf{a}\| := \max_{i=1}^n \|a_i\|$. For a matrix $\mathbf{A} = (A_{i,j})_{i,j}$, the max-norm is taken, i.e. $\|\mathbf{A}\| = \max_{i,j} \|M_{i,j}\|$. The ring expansion factor of $\mathcal{R}$ is defined as $\gamma_\mathcal{R} := \max_{a,b\in\mathcal{R}} \|a \cdot b\| / (\|a\| \cdot \|b\|)$. For power-of-2 and prime-power $\mathcal{R}$, it is known that $\gamma_\mathcal{R} \leq 2\varphi(\rho)$, where φ is Euler's totient function. A set $S \subseteq \mathcal{R}$ is said to be subtractive if $a - b \in \mathcal{R}^\times$ for any distinct $a, b \in S$. For a prime-power $\mathcal{R}$, it is known that $S := \{(\zeta^i - 1)/(\zeta - 1) : i \in [\mathsf{rad}(\rho) - 1]\} \subset \mathcal{R}^\times$ is subtractive, where $\mathsf{rad}(\rho)$ denotes the radical. Note that $\|r\| = 1$ for all $r \in S$.

4.3.2 Lattice Trapdoors

In our constructions based on the (knowledge-)k-R-ISIS assumption, we will make use of lattice trapdoor algorithms. Let η, m, q, β be functions of λ. Let $(\mathsf{TrapGen}, \mathsf{SampD}, \mathsf{SampPre})$ be PPT algorithms parametrised by (η, m, q, β) with the following syntax and properties [GPV08, MP12, GM18]:

- $(\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$ generates a matrix $\mathbf{D} \in \mathcal{R}_q^{\eta\times m}$ and a trapdoor td. The distribution of $\mathbf{D}$ is statistically close to the uniform distribution over $\mathcal{R}_q^{\eta\times m}$.
- $\mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda)$ samples a vector $\mathbf{u} \in \mathcal{R}^m$. For any $(\mathbf{D}, \mathbf{v}) \in \mathcal{R}_q^{\eta\times m} \times \mathcal{R}_q^\eta$ and $\mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda)$ subject to $\mathbf{Du} = \mathbf{v} \bmod q$, it is guaranteed that $\|\mathbf{u}\| \leq \beta$ with overwhelming probability. Furthermore, the following distributions are statistically close:

$$\left\{\begin{array}{l}(\mathbf{D}, \mathbf{u}, \mathbf{v}) : \\ \quad \mathbf{D} \leftarrow \mathcal{R}_q^{\eta\times m} \\ \quad \mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda) \\ \quad \mathbf{v} = \mathbf{Du} \bmod q\end{array}\right\} \quad \text{and} \quad \left\{\begin{array}{l}(\mathbf{D}, \mathbf{u}, \mathbf{v}) : \\ \quad \mathbf{D} \leftarrow \mathcal{R}_q^{\eta\times m} \\ \quad \mathbf{v} \leftarrow \mathcal{R}_q^\eta \\ \quad \mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda) : \mathbf{Du} = \mathbf{v} \bmod q\end{array}\right\}$$

- $\mathbf{u} \leftarrow \mathsf{SampPre}(\mathsf{td}, \mathbf{v})$ inputs a target vector $\mathbf{v} \in \mathcal{R}_q^\eta$ and samples a vector $\mathbf{u} \in \mathcal{R}^m$. For $(\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$, it is guaranteed that $\mathbf{D} \cdot \mathbf{u} = \mathbf{v} \bmod q$ and $\|\mathbf{u}\| \leq \beta$ with overwhelming probability. Furthermore, for any $\mathbf{v} \in \mathcal{R}_q^\eta$, the following distributions are statistically close:

$$\left\{ \begin{array}{l} (\mathbf{D}, \mathbf{u}) : \\ \quad (\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda) \\ \quad \mathbf{u} \leftarrow \mathsf{SampPre}(\mathsf{td}, \mathbf{v}) \end{array} \right\} \quad \text{and} \quad \left\{ \begin{array}{l} (\mathbf{D}, \mathbf{u}) : \\ \quad (\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda) \\ \quad \mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda) : \mathbf{D}\mathbf{u} = \mathbf{v} \bmod q \end{array} \right\}$$

4.3.3 Presumed Hard Problems

The Short Integer Solution (SIS) problem was introduced in the seminal work of Ajtai [Ajt96]. It asks to find a short vector in the kernel of a given random matrix modulo q. In this work, we consider the generalisation of SIS over $\mathcal{R}$ and the k-R-ISIS problem introduced in [ACL+22].

Definition 4.3.1 (R-SIS Assumption)**.** *Let $m, q, \beta^* \in \mathbb{N}$ depend on λ. The Ring-SIS (or R-SIS) problem, denoted $R\text{-}\mathsf{SIS}_{\mathcal{R},m,q,\beta^*}$, is: Given $\mathbf{h} \leftarrow \mathcal{R}_q^m$, find $\mathbf{u} \in \mathcal{R}^m$ such that $0 < \|\mathbf{u}\| \leq \beta^*$ and $\mathbf{h} \cdot \mathbf{u} \equiv \mathbf{0} \bmod q$. We write $\mathsf{Adv}^{\mathsf{r\text{-}sis}}_{\mathcal{R},m,q,\beta^*,\mathcal{A}}$ for the advantage of any algorithm $\mathcal{A}$ in solving $R\text{-}\mathsf{SIS}_{\mathcal{R},\eta,m,q,\beta^*}$. The $R\text{-}\mathsf{SIS}_{\mathcal{R},\eta,m,q,\beta^*}$ assumption states that, for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{r\text{-}sis}}_{\mathcal{R},m,q,\beta^*,\mathcal{A}} \leq \mathsf{negl}(\lambda)$.*

We state a streamlined version of the (knowledge) k-R-ISIS[5] assumptions defined in [ACL+22] with two main changes: 1. To improve readability, our definitions of the assumptions do not impose admissibility constraints on parameters. Instead, we mention these admissibility parameters separately outside of the definitions. 2. We assume that all preimages $\mathbf{u}_g$ given to the adversary are sampled from the same distribution conditioned on different constraints. The original definitions [ACL+22] are more general in that they allow a different distribution per constraint.

Definition 4.3.2 (k-R-ISIS Assumptions)**.** *Let $\eta, m, q, \beta, \beta^* \in \mathbb{N}$, $\mathcal{G} \cup \{ g^* \}$ be a set of w-variate Laurent monomials, $\mathcal{T} \sim \mathcal{R}_q^\eta$, and $\mathcal{D} \sim \mathcal{R}^m$, all dependent on λ. Write $\mathsf{pp} := (\mathcal{R}, \eta, m, w, q, \beta, \beta^*, \mathcal{G}, g^*, \mathcal{D}, \mathcal{T})$. The k-R-$\mathsf{ISIS}_{\mathsf{pp}}$ assumption states that, for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{k\text{-}r\text{-}isis}}_{\mathsf{pp},\mathcal{A}} \leq \mathsf{negl}$, where $\mathsf{Adv}^{\mathsf{k\text{-}r\text{-}isis}}_{\mathsf{pp},\mathcal{A}} :=$*

$$\Pr\left[\begin{array}{l} \mathbf{D} \cdot \mathbf{u}_{g^*} \equiv \mathbf{t} \cdot s^* \cdot g^*(\mathbf{v}) \bmod q \\ \wedge\ 0 < \|(\mathbf{u}_{g^*}, s^*)\| \leq \beta^* \end{array} \middle| \begin{array}{l} \mathbf{D} \leftarrow \mathcal{R}_q^{\eta \times m};\ \mathbf{t} \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w \\ \mathbf{u}_g \leftarrow \mathcal{D} : \mathbf{D} \cdot \mathbf{u}_g = \mathbf{t} \cdot g(\mathbf{v}) \bmod q,\ \forall\ g \in \mathcal{G} \\ (s^*, \mathbf{u}_{g^*}) \leftarrow \mathcal{A}\left(\mathbf{D}, \mathbf{t}, \mathbf{v}, \{ \mathbf{u}_g \}_{g \in \mathcal{G}}\right) \end{array} \right].$$

Individual parameters are omitted when they are clear from the context.

[5] In [ACL+22], the assumptions over modules were separately called (knowledge-)k-M-ISIS.

Definition 4.3.3 (Knowledge k-R-ISIS Assumptions)**.** *Let $\eta, m, q, \alpha^*, \beta, \beta^* \in \mathbb{N}$, $\mathcal{G}$ be a set of w-variate Laurent monomials, $\mathcal{T} \sim \mathcal{R}_q^\eta$, and $\mathcal{D} \sim \mathcal{R}^m$, all dependent on λ. Let $\mathcal{Z}$ be a PPT auxiliary input generator. Write $\mathsf{pp} \coloneqq (\mathcal{R}, \eta, m, w, q, \alpha^*, \beta, \beta^*, \mathcal{G}, \mathcal{D}, \mathcal{T}, \mathcal{Z})$. The knowledge k-R-$\mathsf{ISIS}_{\mathsf{pp}}$ assumption states that for any PPT adversary $\mathcal{A}$ there exists a PPT extractor $\mathcal{E}_\mathcal{A}$ such that $\mathsf{Adv}^{\mathsf{k\text{-}r\text{-}isis}}_{\mathsf{pp},\mathcal{A}} \leq \mathsf{negl}(\lambda)$, where $\mathsf{Adv}^{\mathsf{k\text{-}r\text{-}isis}}_{\mathsf{pp},\mathcal{A}} \coloneqq$*

$$\Pr\left[\begin{array}{l} \mathbf{D}\cdot\mathbf{u} \equiv \mathbf{t}\cdot c \bmod q \\ \wedge\ 0 < \|\mathbf{u}\| \leq \beta^* \\ \wedge\ \neg\left(\begin{array}{l} c \equiv \sum\limits_{g\in\mathcal{G}} x_g \cdot g(\mathbf{v}) \bmod q \\ \wedge\ \left\|(x_g)_{g\in\mathcal{G}}\right\| \leq \alpha^* \end{array}\right) \end{array} \middle| \begin{array}{l} \mathbf{D}\leftarrow\mathcal{R}_q^{\eta\times m};\ \mathbf{t}\leftarrow\mathcal{T};\ \mathbf{v}\leftarrow(\mathcal{R}_q^\times)^w \\ \mathbf{u}_g\leftarrow\mathcal{D} : \mathbf{D}\cdot\mathbf{u}_g = \mathbf{t}\cdot g(\mathbf{v}) \bmod q,\ \forall\ g\in\mathcal{G} \\ \mathsf{pp} \coloneqq (\mathbf{D}, \mathbf{t}, \mathbf{v}, \{\,\mathbf{u}_g\,\}_{g\in\mathcal{G}});\ \mathsf{aux}\leftarrow\mathcal{Z}(\mathsf{pp}) \\ \left((c,\mathbf{u}),(x_g)_{g\in\mathcal{G}}\right)\leftarrow(\mathcal{A}\|\mathcal{E}_\mathcal{A})\,(\mathsf{pp},\mathsf{aux}) \end{array}\right]$$

where $(\mathcal{A}\|\mathcal{E}_\mathcal{A})$ means that $\mathcal{A}$ and $\mathcal{E}_\mathcal{A}$ are run on the same input including the randomness, and $(c, \mathbf{u})$ and $(x_g)_{g\in\mathcal{G}}$ are the outputs of $\mathcal{A}$ and $\mathcal{E}_\mathcal{A}$ respectively. Individual parameters are omitted when they are clear from the context.

For both assumptions to be meaningful, we always consider $m > \eta$.[6] For non-triviality, we want $g^* \notin \mathcal{G}$ and $\mathbf{t} \neq 0$ with overwhelming probability. To avoid complications of giving the adversary short vectors in the kernel of $\mathbf{D}$, we do not consider the case where $\mathcal{G}$ is a multiset – all monomials in $\mathcal{G}$ are distinct.[7] To avoid SIS attacks in the image space, we want $1/|\mathcal{R}_q^\times| = \mathsf{negl}(\lambda)$.

For the knowledge assumption to be plausible, we would like that $\alpha^* \geq \beta^*$, and for $\mathbf{t}\leftarrow\mathcal{T}$, $1/|\langle\mathbf{t}\rangle| = \mathsf{negl}(\lambda)$ and $|\langle\mathbf{t}\rangle|/|\mathcal{R}_q^\eta| = \mathsf{negl}(\lambda)$ with overwhelming probability. Furthermore, to avoid easy instances of ideal-SVP (relevant when $\eta = 1$), we would like the problem of finding short elements in $\{\, s \in \mathcal{R} : \mathbf{t}\cdot s = \mathbf{0} \bmod q \,\}$ to be hard.

4.3.4 Argument Systems

We recall the definition of argument systems which allow a prover to convince a verifier that a relation is satisfiable. Formally, we define a (family of) relation(s) $\Psi(= (\Psi_\lambda)_{\lambda\in\mathbb{N}})$ to be polynomial-time-decidable triples of the form $(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit})$, corresponding to the public parameters of the argument system, the statement, and the witness respectively. We consider a statement $\mathsf{stmt} = (\mathsf{stmt}_{\mathtt{off}}, \mathsf{stmt}_{\mathtt{on}})$ to consist an offline part $\mathsf{stmt}_{\mathtt{off}}$ which is potentially preprocessable and an online part $\mathsf{stmt}_{\mathtt{on}}$. For any fixed public parameters pp, we define the (sub-)relation $\Psi_{\mathsf{pp}} \coloneqq \{\,(\mathsf{stmt}, \mathsf{wit}) : (\mathsf{pp}, \mathsf{stmt}, \mathsf{wit}) \in \Psi\,\}$ and the corresponding language $\mathcal{L}_{\mathsf{pp}} \coloneqq \{\,\mathsf{stmt} : \exists\ \mathsf{wit},\ (\mathsf{stmt}, \mathsf{wit}) \in \Psi_{\mathsf{pp}}\,\}$. We focus on relations where the public parameters pp can be efficiently generated, and denote such a generator by Gen_Ψ. We suppress pp when it is the empty string.

[6] In [ACL+22], m is considered to be large enough so that the leftover hash lemma holds. However, smaller m only makes the problems harder.

[7] In [ACL+22, Definition 22], monomials in $\mathcal{G}$ and g^* are further required to be independent of $\mathcal{R}$. We discuss in Section 4.4.2 why we believe that this restriction can be lifted.

Definition 4.3.4 (Arguments)**.** *A (preprocessing) argument system consists of PPT algorithms* $(\mathsf{Setup}, \mathsf{PreVerify})$ *and PPT interactive algorithms* $(\mathsf{Prove}, \mathsf{Verify})$ *with the following syntax:*

- $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp})$*: Input some public parameters* pp *and generate a common reference string* crs*.*
- $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$*: Preprocess the statement* $\mathsf{stmt}_{\mathtt{off}}$*. Systems not supporting preprocessing are captured by having a trivial preverification, i.e.* $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} = (\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$*.*
- $(\mathsf{tx}, b) \leftarrow \langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}) \rangle$*: An interactive protocol where the prover tries to convince the verifier about the statement* stmt*. The protocol produces a transcript* tx *and ends with the verifier outputting a bit* $b \in \{0,1\}$*. The transcript* tx *is suppressed from the output when it is not needed. In the case where the protocol is non-interactive, i.e. the prover sends a single message, then we split the protocol into two PPT algorithms* $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$ *and* $b \leftarrow \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}, \pi)$*, where* π *is referred to as a proof.*

Definition 4.3.5 (Completeness)**.** *An argument system* Π *is said to be complete for* Ψ *if for all adversaries* $\mathcal{A}$

$$\Pr\left[\begin{array}{l|l} (\mathsf{stmt}, \mathsf{wit}) \in \Psi_{\mathsf{pp}} & \mathsf{pp} \leftarrow \mathsf{Gen}_\Psi(1^\lambda);\ \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp}) \\ \wedge\ b = 0 & (\mathsf{stmt}, \mathsf{wit}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{crs}) \\ & \mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}}) \\ & b \leftarrow \langle \mathcal{P}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathcal{V}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}) \rangle \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Definition 4.3.6 (Special Soundness)**.** *An argument system* Π *is said to be public-coin if each message sent by* $\mathcal{V}$ *is sampled from a public distribution independent of the messages sent by* Prove*. A transcript* tx *is said to be accepting for* $(\mathsf{pp}, \mathsf{stmt})$ *if* $(\mathsf{tx}, 1)$ *is in the output space of* $\langle \mathcal{P}, \mathcal{V}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}) \rangle$ *where* $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \in \mathsf{PreVerify}(\mathsf{Setup}(1^\lambda, \mathsf{pp}), \mathsf{stmt})$*. Suppose* $\mathcal{V}$ *sends* ℓ *messages throughout the execution of* $\langle \mathcal{P}, \mathcal{V} \rangle$*. A tree* T *is said to be a* $(k_1, \ldots, k_\ell)$*-tree of accepting transcripts for* $(\mathsf{pp}, \mathsf{stmt})$ *if it is of (node-)depth* $(\ell + 1)$*, each node is labelled by a prover message, each depth-*i *node has exactly* k_i *children each connected by an edge labelled by a distinct verifier message, and the labels on each root-to-leaf path give an accepting transcript for* $(\mathsf{pp}, \mathsf{stmt})$*. The argument system* Π *is said to be* $(k_1, \ldots, k_\ell)$*-special-sound for* Ψ *if there exists a polynomial-time extractor* $\mathcal{E}$ *which on input a* $(k_1, \ldots, k_\ell)$*-tree of accepting transcripts for* $(\mathsf{pp}, \mathsf{stmt})$ *outputs* wit^* *such that* $(\mathsf{stmt}, \mathsf{wit}^*) \in \Psi_{\mathsf{pp}}$*.*

Definition 4.3.7 (Knowledge Soundness)**.** *Let* $\kappa = \kappa(\lambda)$ *denote the knowledge error. An argument system* Π *is said to be* κ*-knowledge-sound for* Ψ *if for all PPT* $\mathcal{P}^*$ *there exists an expected polynomial-time extractor* $\mathcal{E}_{\mathcal{P}^*}$ *such that for all PPT adversaries* $\mathcal{A}$ *the*

following is at most κ:

$$\Pr\left[\begin{array}{l|l} (\mathsf{stmt},\mathsf{wit}^*)\notin\Psi_{\mathsf{pp}} & \mathsf{pp}\leftarrow\mathsf{Gen}_\Psi(1^\lambda);\ \mathsf{crs}\leftarrow\mathsf{Setup}(1^\lambda,\mathsf{pp}) \\ \wedge\ b=1 & (\mathsf{stmt},\mathsf{wit})\leftarrow\mathcal{A}(\mathsf{pp},\mathsf{crs}) \\ & \mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}\leftarrow\mathsf{PreVerify}(\mathsf{crs},\mathsf{stmt}_{\mathtt{off}}) \\ & (\mathsf{wit}^*,b)\leftarrow\langle(\mathcal{P}^*|\mathcal{E}_{\mathcal{P}^*})(\mathsf{crs},\mathsf{stmt},\mathsf{wit}),\mathcal{V}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}},\mathsf{stmt}_{\mathtt{on}})\rangle \end{array}\right]$$

The argument system Π is said to be knowledge-sound for Ψ if it is κ-knowledge-sound for Ψ for some $\kappa=\mathsf{negl}(\lambda)$.

It is known that a parallel-repetition of a $(k_1,\ldots,k_\ell)$-special-sound protocol yields a knowledge-sound protocol [AF22].

Note that it is common for lattice-based argument systems to have a "soundness gap": They are complete for a relation Ψ, but special- or knowledge-sound for a relaxed relation $\Psi'\supseteq\Psi$, i.e. the extracted witness wit^* for $(\mathsf{pp},\mathsf{stmt})$ may not satisfy $(\mathsf{stmt},\mathsf{wit}^*)\in\Psi_{\mathsf{pp}}$ but only $(\mathsf{stmt},\mathsf{wit}^*)\in\Psi'_{\mathsf{pp}}$.

Definition 4.3.8 (Succinctness)**.** *An argument system Π is said to have succinct proofs (resp. succinct verifier) for Ψ if for any $\mathsf{pp}\in\mathsf{Gen}_\Psi(1^\lambda)$, $\mathsf{crs}\in\mathsf{Setup}(1^\lambda,\mathsf{pp})$, $(\mathsf{stmt},\mathsf{wit})\in\Psi_{\mathsf{pp}}$, $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}\in\mathsf{PreVerify}(\mathsf{crs},\mathsf{stmt}_{\mathtt{off}})$, the communication complexity of $\langle\mathsf{Prove}(\mathsf{crs},\mathsf{stmt},\mathsf{wit}),\mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}},\mathsf{stmt}_{\mathtt{on}})\rangle$ (resp. computation complexity of $\mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}},\mathsf{stmt}_{\mathtt{on}})$) is $\mathsf{polylog}(|\mathsf{stmt}|+|\mathsf{wit}|)\cdot\mathsf{poly}(\lambda)$ where the $\mathsf{poly}(\lambda)$ factor is independent of $|\mathsf{stmt}|$ and $|\mathsf{wit}|$.*

Argument systems which are succinct, non-interactive, and knowledge-sound are known as succinct non-interactive arguments of knowledge (SNARK). Arguments whose soundness holds even against adversaries given the randomness of Setup are said to have transparent setups.

4.4 Vanishing Short Integer Solutions

In this section, we formalise the vanishing-SIS problems and assumptions, and discuss their relations with existing problems and assumptions. We also discuss the properties of the collision-resistant hash functions obtained immediately from the vanishing-SIS assumptions.

4.4.1 Definition

Definition 4.4.1 (Vanishing-SIS)**.** *Let $n,d,w,q,\beta\in\mathbb{N}$ and $\mathcal{G}$, a set of w-variate (Laurent) monomials of individual degree at most d, be functions of λ. The $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},n,q,\beta}$ problem is the following: Given a set $V=\{\mathbf{v}_i\}_{i=1}^n\in(\mathcal{R}_q^\times)^w$ of n uniformly random*

points in $(\mathcal{R}_q^\times)^w$*, find a non-zero polynomial* $p \in \mathcal{R}[X_1, \ldots, X_w]$ *with monomial support*[8] *over* $\mathcal{G}$ *such that*

$$\forall i \in [n], \qquad p(\mathbf{v}_i) = 0 \bmod q \qquad \text{and} \qquad \|p\| \leq \beta$$

where $\|p\|$ *is the maximum of the norm of the coefficients of* p*. The* $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},n,q,\beta}$ *assumption states that, for any PPT adversary* $\mathcal{A}$*, the probability of* $\mathcal{A}$ *solving a uniformly random instance of* $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},n,q,\beta}$ *is negligible in* λ*. Individual parameters are omitted from the subscript when they are clear from the context. If* $\mathcal{G}$ *is the set of all* w*-variate (Laurent) monomials of individual degree at most* d*, we denote the problem by* $\mathsf{vSIS}_{\mathcal{R},d,w,n,q,\beta}$*. To emphasise certain parameters, e.g.* $n = n^*$ *and* $w = w^*$*, we sometimes write* $\mathsf{vSIS}_{(n,w)=(n^*,w^*)}$*.*

Another way to phrase the problem, borrowing terminologies from algebraic geometry, is that it asks to find an element of bounded norm and degree in the ideal $\mathcal{I}(V)$ of polynomials vanishing at the set of points V. Clearly, the subset of bounded-degree polynomials in $\mathcal{I}(V)$ forms a (module) lattice. Therefore a vanishing-SIS problem can also be seen as an average-case approximate shortest vector problem (SVP) over such lattices.[9]

The connection of the vanishing-SIS problem to the standard SIS problem stems from the following simple observation: If we interpret the coefficients of a solution p as a vector $\mathbf{p}$, and write the relation in matrix form, we obtain

$$\begin{pmatrix} 1 & v_{1,1} & \ldots & v_{1,w} & \ldots & \prod_{j=1}^{w} v_{1,j}^{e_j} & \ldots & \prod_{j=1}^{w} v_{1,j}^{d} \\ 1 & v_{2,1} & \ldots & v_{2,w} & \ldots & \prod_{j=1}^{w} v_{2,j}^{e_j} & \ldots & \prod_{j=1}^{w} v_{2,j}^{d} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & v_{n,1} & \ldots & v_{n,w} & \ldots & \prod_{j=1}^{w} v_{n,j}^{e_j} & \ldots & \prod_{j=1}^{w} v_{n,j}^{d} \end{pmatrix} \cdot \mathbf{p} = \mathbf{0} \bmod q \quad \text{and} \quad \|\mathbf{p}\| \leq \beta,$$

a SIS relation with respect to a (Vandermonde-like) structured matrix.

Note that since $v_{i,j} \in \mathcal{R}_q^\times$ for all i and j, it is not important for p to be a polynomial with only non-negative powers. Laurent polynomials can be captured scaling the each i-th row of the matrix by $\prod_{j=1}^{w} v_{i,j}^{-e_j}$ for any desired powers $(e_1, \ldots, e_w) \in \mathbb{Z}^w$. In fact, using the matrix formulation, the scaling factors for each row could be different.

It is easy to observe that the vanishing-SIS assumption is implied by the k-R-ISIS assumption with related parameters. In Section 4.5, we discuss this implication in more detail, show that the converse holds conditioned on a related knowledge-k-R-ISIS assumption, and explore the connections of vanishing-SIS to more established assumptions, i.e. NTRU and RingLWE.

[8] e.g. the monomial support of $3X_1X_2 + 2X_2^2 + 1$ is $\{ X_1X_2, X_2^2, 1 \}$

[9] Interestingly, after restricting to a bounded-degree subset, we no longer have an ideal. Therefore this approximate SVP problem is not over ideal-lattices.

4.4.2 On Choice of Parameters

On the modulus q. Note that, for some (preferable) parameters settings, it is important for $q > d$ for the vSIS assumption to be plausible. Indeed, for example, if q is prime and is such that $q\mathcal{R}$ splits completely into $\varphi(\rho)$ ideals, then we have $v^{q-1} - 1 = 0 \bmod q$ for any $v \in \mathcal{R}$. This gives rise to trivial solutions, e.g. $p(X) = X^{q-1} - 1$, to the vSIS problem.

On the space of V. It is also important for the set of points V to be chosen over $\mathcal{R}_q^\times$ instead of $\mathcal{R}_q$. For example, consider a power-of-2 $\mathcal{R}$ and $q = 2^\ell$. The ideal $q\mathcal{R}$ splits into $q\mathcal{R} = \mathcal{I}^{\ell\cdot\varphi(\rho)}$ for some ideal $\mathcal{I}$ of (algebraic) norm $\mathcal{N}(\mathcal{I}) = 2$. Therefore, with probability $1/2$, a random element $v \leftarrow \mathcal{R}_q$ satisfies $v = 0 \bmod \mathcal{I}$ and hence $v^{\ell\cdot\varphi(\rho)} = 0 \bmod q$. This means that $p(X) = X^{\ell\cdot\phi(\rho)}$ is a solution to any vanishing-SIS over $\mathcal{R}_q$ if instances were sampled from $\mathcal{R}_q$.[10]

On the cardinality $|\mathcal{R}_q^\times|$. It is crucial that the cardinality $|\mathcal{R}_q^\times|$ is large enough so that $1/|\mathcal{R}_q^\times| = \mathsf{negl}$. Suppose not, then there might exist small $e \in \mathbb{N}$ such that $\{ v, v^2, \ldots, v^e \}$ contains a short element modulo q. Note that the set of elements in $\mathcal{R}$ of norm at most β has cardinality $(2\beta+1)^{\varphi(\rho)}$. If we heuristically model the multiplication-by-v map $a \mapsto a \cdot v \bmod q$ as a random permutation for $v \leftarrow \mathcal{R}_q^\times$, and if $\mathcal{R}_q^\times$ is large enough, we have some confidence to believe that small powers of v modulo q will not be short.

In general, it appears that $|\mathcal{R}_q^\times|$ is usually quite close to $q^{\varphi(\rho)}$. We calculate this cardinality for some specific choices of q and $\mathcal{R}$. For $q = 2^\ell$ and ρ being a power of 2, we have $|\mathcal{R}_q^\times| = q^{\varphi(\rho)}/2$. For arbitrary $\mathcal{R}$ and prime $q = 1 \bmod \varphi(\rho)$, we have $|\mathcal{R}_q^\times| = (q-1)^{\varphi(\rho)}$. In either case, if $\beta \le q/4$, we have $\Pr[\|x\| \le \beta \mid x \leftarrow \mathcal{R}_q^\times] < 2^{-\varphi(\rho)}$ which is negligible in ρ.

4.4.3 A Family of Hash Functions with Short Keys

Similar to the standard SIS-based hash function, the vanishing-SIS assumption immediately implies the existence of a collision-resistant hash function, except that in this case the keys are very small, and could potentially be *logarithmic* in the message size. Furthermore, the hash function satisfies many desirable properties, such as (approximate) ring homomorphism.

In more detail, for any set of points $V = \{ \mathbf{v}_i \}_{i=1}^n \subseteq ((\mathcal{R}_q^\times)^w)^n$, define

$$\mathcal{H}_V : \mathcal{R}_\beta^{(d+1)w} \to \mathcal{R}_q^n,\ \mathcal{H}_V(p) = (p(\mathbf{v}_1), \ldots, p(\mathbf{v}_n)) \bmod q$$

where an input $\mathbf{p} \in \mathcal{R}_\beta^{(d+1)w}$ is interpreted, for example, as a polynomial $p \in \mathcal{R}_\beta[X_1, \ldots, X_w]$ of individual degree at most d.

[10] We believe that this is the reason why $\mathcal{G}$ was restricted to be independent of $\mathcal{R}$ in the definition of "k-R-ISIS-admissible" parameters in [ACL+22, Definition 22]. However, since [ACL+22, Definition 23] also restricts $\mathbf{v} \in (\mathcal{R}_q^\times)^w$, the restriction on $\mathcal{G}$ appears to be redundant.

It is easy to show that this function is collision resistant by observing that $\mathcal{H}_V(p) = \mathcal{H}_V(p')$ implies

$$\forall i \in [n],\ (p - p')(\mathbf{v}_i) = 0 \bmod q \qquad \text{and} \qquad \|p - p'\| \leq \beta,$$

i.e. $p - p'$ is a solution to the vSIS instance V.

Observe that each hash function can be described by a key of size $n \cdot w \log q$ bits, and can hash messages of length $(d+1) \cdot w \cdot \log \beta$ bits to $n \cdot \log q$ bits, where n and w could be as small as 1. As discussed in Section 4.4.1, for the vSIS assumption to be plausible for the case where q fully splits, which is desirable for efficiency, it is necessary that $q > d$. For $q = O(d)$ and $n, w, \beta = \mathsf{poly}(\lambda)$, the key size and the message length are $O_\lambda(\log d)$ and $O_\lambda(d)$ respectively.

Similar to the standard SIS-based hash function, $\mathcal{H}_V$ is almost linearly homomorphic in the sense that

$$\mathcal{H}_V(p) + \mathcal{H}_V(p') = \mathcal{H}_V(p + p') \bmod q \qquad \text{and} \qquad \|p + p'\| \leq \|p\| + \|p'\|.$$

Different from the standard SIS-based hash function, however, is that $\mathcal{H}_V$ is also almost multiplicatively homomorphic in the sense that

$$\mathcal{H}_V(p) \cdot \mathcal{H}_V(p') = \mathcal{H}_V(p \cdot p') \bmod q \qquad \text{and} \qquad \|p \cdot p'\| \leq (d+1)^w \cdot \|p\| \cdot \|p'\| \cdot \gamma_{\mathcal{R}},$$

with multiplications taken over $\mathcal{R}_q$ and $\mathcal{R}[\mathbf{X}]$ respectively.

For our purpose of construction linear-time succinct arguments, the univariate case (i.e. $w = 1$) is the most interesting due to the exponential dependency of various parameters on w. Moreover, we notice that if $p_0(X)$ and $p_1(X)$ encode the vectors $\mathbf{p}_0$ and $\mathbf{p}_1$ respectively as their coefficients, then the product polynomial $p(X) \cdot p(X^{-1})$ has norm at most $\|\mathbf{p}_0\| \cdot \|\mathbf{p}_1\| \cdot \gamma_{\mathcal{R}}$, and its constant term encodes the inner product $\langle \mathbf{p}_0, \mathbf{p}_1 \rangle$.

4.5 Relating Vanishing-SIS and other Assumptions

In the following, we show that the vanishing-SIS assumption is implied by, and tightly related to, the k-R-ISIS assumption. We also explore the connections between the vanishing-SIS, NTRU, and RingLWE assumptions.

4.5.1 Relations with k-R-ISIS

We discuss how the vSIS assumption relates to the k-R-ISIS family of assumptions defined in [ACL+22]. We show implications in both directions that hold in different parameter regimes.

k-R-ISIS $\implies$ vSIS. We show that $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\},\alpha}$ is no easier than $k\text{-}R\text{-}\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$ with $\beta^* = |\mathcal{G}| \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}$. Assuming that we have a solver for $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\},\alpha}$ that outputs a polynomial p with

$$p(\mathbf{v}) = \sum_{g\in\mathcal{G}} p_g \cdot g(\mathbf{v}) + p_{g^*} \cdot g^*(\mathbf{v}) = 0 \bmod q,$$

we can construct an algorithm solving $k\text{-}R\text{-}\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$ as follows. The algorithm is given as input $\left(\mathbf{D}, \mathbf{t}, \mathbf{v}, \{\,\mathbf{u}_g\,\}_{g\in\mathcal{G}}\right)$. It runs the $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\},\alpha}$ solver on $\mathbf{v}$ to obtain p, and returns

$$\mathbf{u}_{g^*} = \sum_{g\in\mathcal{G}} p_g \cdot \mathbf{u}_g \text{ and } s^* = -p_{g^*}.$$

Note that this is a valid solution for $k\text{-}R\text{-}\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$ since

$$\mathbf{D}\cdot\mathbf{u}_{g^*} = \mathbf{D}\cdot\sum_{g\in\mathcal{G}} p_g\cdot\mathbf{u}_g = \sum_{g\in\mathcal{G}} p_g\cdot g(\mathbf{v}) = -p_{g^*}\cdot g^*(\mathbf{v}) \bmod q$$

and furthermore, by assumption, we have that $\|p\| \le \alpha$ and thus

$$\|\mathbf{u}_{g^*}\| \le |\mathcal{G}|\cdot\alpha\cdot\beta\cdot\gamma_{\mathcal{R}} = \beta^* \qquad \text{and} \qquad \|p_{g^*}\| \le \alpha < \beta^*.$$

Knowledge k-R-ISIS and vSIS $\implies$ k-R-ISIS. We show that, conditioned on knowledge $k\text{-}R\text{-}\mathsf{ISIS}_{\mathcal{G},\alpha^*,\beta,\beta^*}$, the hardness of $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\},\alpha^*}$ implies that of $k\text{-}R\text{-}\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$, for $\alpha^* \ge \beta^*$. At first glance, it may appear strange that we are using the knowledge version of k-R-ISIS to prove k-R-ISIS itself. Nevertheless the statement is meaningful, since knowledge k-R-ISIS is not a computational problem, but rather an assumption about the attacker itself. In some sense, this statement shows that vSIS is the underlying computational assumption that connects k-R-ISIS and knowledge k-R-ISIS. We sketch the reduction in the following. Assume that we are given a $k\text{-}R\text{-}\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$ solver that, on input $\left(\mathbf{D}, \mathbf{t}, \mathbf{v}, \{\,\mathbf{u}_g\,\}_{g\in\mathcal{G}}\right)$, outputs $(\mathbf{u}_{g^*}, s^*)$ such that

$$\mathbf{D}\cdot\mathbf{u}_{g^*} = \mathbf{t}\cdot s^*\cdot g^*(\mathbf{v}) \qquad \text{and} \qquad \|(\mathbf{u}_{g^*}, s^*)\| \le \beta^* \le \alpha^*.$$

By knowledge $k\text{-}R\text{-}\mathsf{ISIS}_{\mathcal{G},\alpha^*,\beta,\beta^*}$, there exists an extractor that returns $\{\,x_g\,\}_{g\in\mathcal{G}}$ such that

$$s^*\cdot g^*(\mathbf{v}) = \sum_{g\in\mathcal{G}} x_g\cdot g(\mathbf{v}) \qquad \text{and} \qquad \|x_g\| \le \alpha^*.$$

It follows that $p = \sum_{g\in\mathcal{G}} x_g\cdot g - s^*\cdot g^*$ is a valid solution for $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\}}$ since

$$\sum_{g\in\mathcal{G}} x_g\cdot\mathbf{u}_g - s^*\cdot g^*(\mathbf{v}) = p(\mathbf{v}) = 0 \bmod q \qquad \text{and} \qquad \|p\| \le \alpha^*.$$

4.5.2 Relations with NTRU

In the following, we study the relation between vSIS and the search NTRU assumption, conditioned on the decision NTRU assumption. Recall that the decision NTRU assumption states that the "NTRU distribution", i.e. that of $h = f/g \bmod q \in \mathcal{R}_q^\times$ where $f, g \leftarrow \mathcal{R}$ are random short elements, is indistinguishable from the uniform distribution over $\mathcal{R}_q^\times$. From the decision NTRU assumption, we immediately have that the distribution of vSIS instances is indistinguishable from a modified version where each entry of each point $\mathbf{v} \in V$ is sampled from the NTRU distribution instead of uniformly from $\mathcal{R}_q^\times$. In the following, we refer to this modified version of vSIS as NTRU-vSIS.

Univariate NTRU-vSIS $\implies$ Search NTRU.

First, we make a simple observation that NTRU-vSIS generalises search NTRU. The search NTRU problem is the following: Given h sampled from the NTRU distribution, find short f', g' such that that $g'h + f' = 0 \bmod q$, i.e. find a degree-1 polynomial p with short coefficients which vanishes at h modulo q. Clearly, a search NTRU solver also solves NTRU-$\mathsf{vSIS}_{n=1,\mathcal{G}}$ if $\{1, X\} \subseteq \mathcal{G}$.

Solution Space.

The search NTRU problem can be viewed (see e.g. [PS21]) as the problem of finding a short vector spanned by $\begin{pmatrix} q & -h \\ & 1 \end{pmatrix}$. Similarly, the $\mathsf{vSIS}_{(n,w)=(1,1)}$ problem can be viewed as the problem of finding a short vector in the rank-$(d+1)$ module lattice spanned by $\begin{pmatrix} q & \begin{bmatrix} -v \\ 1 \end{bmatrix}_{\searrow d} \end{pmatrix}$. In [PS21], Pellet-Mary and Stehlé showed that all solutions to a search NTRU problem lie in a unique rank-1 submodule of the module-lattice spanned by $(-f, g)^\intercal$.[11] Similarly, we can show that, for large enough q (exponential in d), all solutions to an NTRU-$\mathsf{vSIS}_{(n,w)=(1,1),d}$ problem lie in a unique rank-d submodule. The argument roughly goes as follows.

Consider $v = f/g \bmod q$ where $\|f\|, \|g\| \leq \alpha$, and let $\mathbf{p}$ be a solution to the NTRU-$\mathsf{vSIS}_{(n,w)=(1,1),d}$ instance v. We have $\sum_{j=0}^{d} p_j \cdot v^j = 0 \bmod q$ or equivalently $\sum_{j=0}^{d} p_j \cdot f^j \cdot g^{d-j} = 0 \bmod q$. Assuming that $q > 2 \cdot (d+1) \cdot \alpha^d \cdot \beta \cdot \gamma_{\mathcal{R}}^d$, we have $\sum_{j=0}^{d} p_j \cdot f^j \cdot g^{d-j} = 0$, with arithmetic done over $\mathcal{K}$. In other words, the solution lies in the kernel of $(g^d, f \cdot g^{d-1}, \ldots, f^d)$ for which a basis is given by $\begin{bmatrix} -f \\ g \end{bmatrix}_{\searrow d}$.

[11] Recovering this submodule (represented by a possibly long vector) was formalised as the $\mathrm{NTRU}_{\mathrm{mod}}$ problem in [PS21]. This variant of the search NTRU problem is trivially not harder than the standard variant, and [PS21] gave a reduction from the decision NTRU problem.

Decision NTRU + Worst-Case $\implies$ Average-Case.

The $\mathsf{vSIS}_{n=1}$ problem admits a worst-case to average-case reduction, conditioned on the hardness of decision NTRU. Note that this reduction produces a solution of norm exponential in d and w. In the following, we sketch the reduction.

Let $\mathbf{v}^*$ be any fixed $\mathsf{vSIS}_{n=1,\beta^*}$ instance for some β^* to be specified later. For each $j \in [w]$, sample an NTRU element $h_j = f_j/g_j \bmod q$ where $\|f_j\|, \|g_j\| \leq \alpha$ for all j. Define $\mathbf{v}$ where $v_j := v_j^* \cdot h_j \bmod q$. Note that $\mathbf{v} \circ \mathbf{g} = \mathbf{v}^* \circ \mathbf{f}$ where $\circ$ denotes the Hadamard product. By the decision NTRU assumption, $\mathbf{v}$ is indistinguishable from a random $\mathsf{vSIS}_{n=1,\beta}$ instance. Suppose p is a solution to the vSIS_β instance $\mathbf{v}$, i.e. $p(\mathbf{v}) = 0 \bmod q$ and $\|p\| \leq \beta$, then

$$\prod_{j=1}^{w} g_j^d \cdot p(\mathbf{v}) = \prod_{j=1}^{w} g_j^d \cdot p\left(\mathbf{v}^* \circ \left(\frac{f_1}{g_1}, \ldots, \frac{f_w}{g_w}\right)\right) = 0 \bmod q.$$

Note that $\prod_{j=1}^{w} g_j^d \cdot p\left(\mathbf{v}^* \circ \left(\frac{f_1}{g_1}, \ldots, \frac{f_w}{g_w}\right)\right)$ can be seen as a polynomial with coefficients in $\mathcal{R}$ evaluated at $\mathbf{v}^*$ (since all denominators are cancelled out). Denote this polynomial by p^*. We have $p^*(\mathbf{v}^*) = 0 \bmod q$. Furthermore, notice that $\|p^*\| \leq \|p\| \cdot \alpha^{d \cdot w} \cdot \gamma_{\mathcal{R}}^{d \cdot w} = \alpha^{d \cdot w} \cdot \beta \cdot \gamma_{\mathcal{R}}^{d \cdot w}$. Therefore p^* is a solution to the $\mathsf{vSIS}_{n=1,\beta^*}$ instance $\mathbf{v}^*$ with $\beta^* = \alpha^{d \cdot w} \cdot \beta \cdot \gamma_{\mathcal{R}}^{d \cdot w}$.

vSIS, NTRU, and RingLWE.

It is clear that the $\mathsf{vSIS}_{n=1}$ problem can be reduced to the vSIS problem (with the same parameters except that n is changed from 1 to an arbitrary polynomial). In the following, we sketch a reduction from the search NTRU problem to the $\mathsf{vSIS}_{(n,w)=(1,1),d}$ problem, conditioned on the hardness of either decision NTRU or RingLWE. We note that this reduction could only work for a certain extreme parameter regime which is not suitable for our application of succinct arguments.

Using Decision NTRU. Given a NTRU-$\mathsf{vSIS}_{(n,w)=(1,1),d,\beta'}$ solver for some β', we would like to find a solution to a random NTRU instance v of norm bounded by some β^*. Interpreting v as an NTRU-$\mathsf{vSIS}_{(n,w)=(1,1),d,\beta}$ instance, using the decision-NTRU rerandomisation technique in the above worst-case to average-case reduction, we can rerandomise v to d $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta'}$ instances v_i for $i \in [d]$, where $\beta := \alpha^d \cdot \beta' \cdot \gamma_{\mathcal{R}}^d$. Let p_i' be a solution of norm β' to the i-th instance v_i. Using the transformation as in the above worst-case to average-case reduction, we can obtain d solutions $(p_i)_{i=1}^d$ to the $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta}$ instance v. Note that each p_i is a degree-d polynomial of norm β. Recursively running the following algorithm produces a degree-1 polynomial of norm $(2\gamma_{\mathcal{R}})^{2^{d-1}-1} \cdot \beta^{2^{d-1}} < (2\beta\gamma_{\mathcal{R}})^{2^{d-1}} =: \beta^*$:

- Input: $L = (p_0, p_1, \ldots, p_{d-1})$, degree-$d$ polynomials each of norm β.
- Output: $L' = (p_1', \ldots, p_{d-1}')$, degree-$(d-1)$ polynomials each of norm $2\beta^2\gamma_{\mathcal{R}}$.
- Procedure:

– For $0 \leq i < d$, let a_i be the coefficient of the degree-d term in p_i.
– Output $L' = (p'_1, \ldots, p'_i)$ where $p'_i = a_0 \cdot p_i - a_i \cdot p_0$.

If we could argue that $(p_1, \ldots, p_d)$ are linearly independent (as polynomials over $\mathcal{K}$) and set $q \gg \beta^*$, then the above gives a solution of norm β^* to the NTRU instance v. In particular, if $\beta^* \ll \sqrt{q}$, then a search NTRU solver would also solve decision NTRU, contradicting the initial assumption that decision NTRU holds. We therefore obtain a reduction from decision NTRU to NTRU-vSIS. Note that for this reduction to work it is necessary to have q being doubly-exponential in the number of monomials $d+1$, which forces d to be constant.

Using RingLWE. Instead of using decision NTRU for rerandomisation, we could use RingLWE by exploiting the fact that we start with a random search NTRU instance v.[12] Specifically, we can rerandomise v to $v_i \coloneqq v \cdot s_i + e_i \bmod q$ for $\|s_i\|, \|e_i\| \leq \alpha$, provided that we reject those v_i which are not invertible. By the (normal-form) RingLWE assumption, each v_i is indistinguishable from a random $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta}$ instance. Suppose p'_i is a solution to the $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta'}$ instance v_i, then

$$p'_i(v_i) = p'_i(v \cdot s_i + e_i),$$

meaning that $p_i(X) = p'_i(X \cdot s_i + e_i)$ is a solution to the $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta}$ instance v, where now $\beta = d \cdot \alpha^d \cdot \beta' \cdot \gamma_{\mathcal{R}}^d$. The rest then follows similar to the reduction using decision NTRU.

4.6 Foldable Structures

We define a family of monomials, polynomials, vectors, and matrices that exhibit "foldable" structures.

Definition 4.6.1 (Foldable Polynomials)**.** *Let $\ell \geq 0$, $k_\ell > 0$, and $k_{\ell-1}, \ldots, k_0 \geq 0$ be integers. A sequence of (monic multivariate Laurent) monomials* $\mathbf{m}$[13] *of length $n = \sum_{i=0}^{\ell} 2^i \cdot k_i$ (where k_i are not necessarily binary) is said to be $(k_0, k_1, \ldots, k_\ell)$-foldable if the following properties are satisfied:*

- $\mathbf{m} = \mathbf{m}_0$ *can be generated from a "seed" $\mathbf{m}_\ell$ and a "generator" $(\ell_i, \mathbf{c}_i, r_i)_{i=0}^{\ell-1}$, where $\mathbf{m}_\ell$ is a sequence of monomials of length k_ℓ, $\mathbf{c}_i$ is a sequence of monomials of length k_i, and ℓ_i, r_i are monomials, in a recursive fashion:*[14]

$$\forall i \in [\ell],\ \mathbf{m}_{i-1}^{\mathsf{T}} \coloneqq \left(\ell_{i-1} \cdot \mathbf{m}_i^{\mathsf{T}} \ \|\ \mathbf{c}_{i-1}^{\mathsf{T}} \ \|\ r_{i-1} \cdot \mathbf{m}_i^{\mathsf{T}}\right).$$

[12] We could not do this in the worst-case to average-case reduction where v was fixed.
[13] That is, each entry of $\mathbf{m}$ is a monic multivariate Laurent monomial.
[14] In the recursive expression, "$\cdot$" denotes the symbolic multiplication of monomials. For example, $X \cdot (X^2, X^3) = (X^3, X^4)$.

- *For all* $i \in \{0, \ldots, \ell\}$, $\mathbf{m}_i$ *consists of distinct monomials.*

We say that $\mathbf{m}$ *is foldable if it is* $(k_0, k_1, \ldots, k_\ell)$*-foldable for some* $(k_0, k_1, \ldots, k_\ell)$*. A foldable polynomial is a polynomial whose supporting monomials can be arranged into a foldable sequence of monomials.*

Note that any sequence of monomials $\mathbf{m}$ of length n is trivially $(0, \ldots, 0, n)$-foldable. However, we are most interested in sequences which are $(k_0, k_1, \ldots, k_\ell)$-foldable for small constants k_i, e.g. $k_i \in \{0, 1, 2\}$, for all $i \in \{0, \ldots, \ell\}$. Below, we state some elementary properties satisfied by foldable monomials.

Lemma 4.6.1. *Let* $\mathbf{m}$ *of length* n *be* $(k_0, \ldots, k_\ell)$*-foldable. Let* $k^* \coloneqq \max_{i=0}^{\ell} k_i$*. It holds that* $\ell \leq \log n < \ell + \log 2 \cdot k^*$.

The proof of Lemma 4.6.1 is deferred to Appendix C.1. The following properties follow immediately from the definition and are stated without proof.

Lemma 4.6.2 (Chaining/Decomposition)**.** *If* $\mathbf{m}$ *is foldable with seed and generator* $(\mathbf{m}', \mathbf{g}')$ *and* $\mathbf{m}'$ *is foldable with seed and generator* $(\mathbf{m}'', \mathbf{g}'')$*, then* $\mathbf{m}$ *is foldable with seed and generator* $(\mathbf{m}'', \mathbf{g}''\|\mathbf{g}')$.

Lemma 4.6.3 (Closure under Hadamard Product)**.** *If* $\mathbf{m}$ *and* $\mathbf{m}'$ *are both* $(k_0, k_1, \ldots, k_\ell)$*-foldable with, where* $\mathbf{m}$ *and* $\mathbf{m}'$ *are supported by disjoint sets of variables and have seeds and generators*

$$(\mathbf{s}, (\ell_i, \mathbf{c}_i, r_i)_{i=0}^{\ell-1}) \qquad \text{and} \qquad (\mathbf{s}', (\ell'_i, \mathbf{c}'_i, r'_i)_{i=0}^{\ell-1})$$

respectively, then the Hadamard product $\mathbf{m} \circ \mathbf{m}'$ *is also* $(k_0, k_1, \ldots, k_\ell)$*-foldable with seed and generator*

$$(\mathbf{s} \circ \mathbf{s}', (\ell_i \cdot \ell'_i, \mathbf{c}_i \circ \mathbf{c}'_i, r_i \cdot r'_i)_{i=0}^{\ell-1}).$$

Next, we extend the definition of foldable monomials and polynomials to that of (block-)foldable vectors and matrices. We then give examples of such objects. The proofs are elementary and are deferred to Appendix C.1.

Definition 4.6.2 (Foldable Vectors and Matrices)**.** *A (row or column) vector* $\mathbf{a} = (a_1, \ldots, a_n)$ *is said to be* $(k_0, k_1, \ldots, k_\ell)$*-foldable if there exists a* $(k_0, k_1, \ldots, k_\ell)$*-foldable sequence of monomials* $\mathbf{m} = (m_1, \ldots, m_n)$ *and a point* $\mathbf{v} \in (\mathcal{R}^\times)^k$ *such that* $a_i = m_i(\mathbf{v})$ *for all* $i \in [n]$*, i.e. the* i*-th entry of* $\mathbf{a}$ *is obtained by evaluating the* i*-th monomial in* $\mathbf{m}$ *at the point* $\mathbf{v}$*. The point* $\mathbf{v}$ *is said to be the evaluation point of* $\mathbf{a}$*. A matrix is said to be foldable if every row of it is foldable with a common evaluation point* $\mathbf{v}$*. A block-matrix* $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_n)$ *where* $\mathsf{ncol}(\mathbf{A}_i) = w$ *for all* $i \in [n]$ *is said to be block-foldable with block-size* w *if, for all* (i, j)*, the vector formed by taking the* (i, j)*-th entry of each of* $(\mathbf{A}_1, \ldots, \mathbf{A}_n)$ *is foldable.*

Lemma 4.6.4 (Power Sequence). *For any $n \in \mathbb{N}$, express n uniquely*[15] *as $n = \sum_{i=0}^{\ell} 2^i \cdot k_i$ with $k_i \in \{1,2\}$ for $i \in \{0,\ldots,\ell\}$. Then for any $v \in \mathcal{R}$, the vector $\mathbf{v}^\intercal = (v, v^2, \ldots, v^n)$. is $(k_0, k_1, \ldots, k_\ell)$-foldable. Generalising, for $w \in \mathbb{N}$, the vector $\mathbf{v}^\intercal = (v, v^2, \ldots, v^{wn})$ is $(k_0, k_1, \ldots, k_\ell)$-block-foldable with block-size w.*

Lemma 4.6.5 (Balanced Power Sequence). *For any $n \in \mathbb{N}$, express n uniquely as $n = \sum_{i=0}^{\ell} 2^i \cdot k_i$ with $k_\ell = 1$ and $k_i \in \{0,1\}$ for all $i \in \{0,\ldots,\ell-1\}$. Then for any $v \in \mathcal{R}$, the following vector is $(0, k_0, k_1, \ldots, k_\ell)$-foldable:*

$$\mathbf{v}^\intercal = (v^{-n}, \ldots, v^{-2}, v^{-1}, v, v^2, \ldots, v^n).$$

Lemma 4.6.6 (Compression Vector). *For any integers $\ell \geq 0$, $k_\ell > 0$ and $k_0, \ldots, k_{\ell-1} \geq 0$, let X_{i,j_i} be independent variables for $i \in \{0,\ldots,\ell\}$ and $j_i \in \{0,\ldots,k_i\}$. The seed and generator*

$$((X_{\ell,1}, \ldots, X_{\ell,k_\ell}), (1, (X_{i,1}, \ldots, X_{i,k_i}), X_{i,0})_{i=0}^{\ell-1})$$

generate a $(k_0, k_1, \ldots, k_\ell)$-foldable sequence of monomials $\mathbf{m}$. Furthermore, let $\mathbf{x} = (x_{i,j})_{i=0,j=1}^{\ell,k_i}$ be a vector over $\mathcal{R}$ with $\|\mathbf{x}\| \leq \alpha$. Let $\mathbf{h} := \mathbf{m}(\mathbf{x})$ be the foldable vector obtained by evaluating $\mathbf{m}$ at $\mathbf{x}$. It holds that $\|\mathbf{h}\| \leq \alpha^{\ell+1} \cdot \gamma_{\mathcal{R}}^{\ell}$.

4.7 Folding Protocols

We state two folding protocols $\Pi_0^{\mathtt{fold}}$ and $\Pi_1^{\mathtt{fold}}$ for bounded-norm satisfiability of (structured) linear relations which respect the foldable structures (Section 4.6) of the matrices and vectors defining the relations. Both protocols have trivial (hence transparent) setup and trivial pre-verification, i.e. $\mathsf{crs} = \Pi_b^{\mathtt{fold}}.\mathsf{Setup}(1^\lambda, \mathsf{pp}) = (1^\lambda, \mathsf{pp})$ and $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} = \Pi_b^{\mathtt{fold}}.\mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}}) = (\mathsf{crs}, \mathsf{stmt})$. We detail below the prove-verify protocols

$$\Pi_b^{\mathtt{fold}}.\langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}) \rangle.$$

4.7.1 Type-0 Linear Relations

Define the relation $\Psi_0^{\mathtt{fold}} = \Psi_0^{\mathtt{fold}}[\mathcal{R}, h_0, h_1, w, n, q_0, q_1, \alpha]$:

$$\Psi_0^{\mathtt{fold}} := \left\{ (\mathsf{pp}, ((\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{y}), \mathbf{z}), \mathbf{x}) : \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow n} \cdot \mathbf{x} = \mathbf{y} \bmod q_0, \quad \text{and} \quad \|\mathbf{x}\| \leq \alpha, \\ \mathbf{C} \cdot \mathbf{x} = \mathbf{z} \bmod q_1, \right\}$$

where $\mathcal{R}$ is a prime-power ring for a prime ≥ 5, $\mathbf{A}, \mathbf{B} \in \mathcal{R}_{q_0}^{h_0 \times w}$, $\mathbf{C} = (\mathbf{C}_1, \ldots, \mathbf{C}_n) \in \mathcal{R}_{q_1}^{h_1 \times wn}$, $\mathbf{y} \in \mathcal{R}_{q_0}^{h_0 \cdot (n+1)}$, $\mathbf{z} \in \mathcal{R}_{q_1}^{h_1}$, and $\mathbf{x} \in \mathcal{R}^{wn}$. Note that the linear constraints consist

[15] Suppose the expression is not unique, let $n = \sum_{i=0}^{\ell} 2^i \cdot k_i = \sum_{i=0}^{\ell} 2^i \cdot k_i'$ with $k_i, k_i' \in \{1,2\}$. Let $d_i = k_i - k_i' \in \{-1,0,1\}$. We have $\sum_{i=0}^{\ell} 2^i \cdot d_i = 0$, which means that $d_0 = 0$ or else the LHS is odd while the RHS is even. Dividing both sides by 2, we get $\sum_{i=0}^{\ell-1} 2^i \cdot d_{i+1} = 0$. By the same argument, we have $d_1 = 0$. Repeating this for all i yields $d_i = 0$ for all $i \in \{0,\ldots,\ell\}$, a contradiction.

of a sparse structured part represented by a block-bidiagonal matrix and a dense part. By default, we suppress all parameters of $\Psi_0^{\mathtt{fold}}$ except those that we highlight. Note that the above constraints are independent of pp, therefore $\Psi_0^{\mathtt{fold}}$ is compatible with any parameter generator Gen. We describe a protocol $\Pi_0^{\mathtt{fold}}$ which is complete for $\Psi_0^{\mathtt{fold}}[\alpha]$ and knowledge sound for $\Psi_0^{\mathtt{fold}}[\alpha^*]$ for some $\alpha^* > \alpha$.

Construction. The protocol $\Pi_0^{\mathtt{fold}}$ is essentially a merge between (the lattice analogue of) Pietrzak's folding protocol [Pie19] and the lattice-based Bulletproofs protocol [BLNS20]. Consider $n > 2$ and let $n' = \lfloor (n-1)/2 \rfloor$. Our protocol hinges on the following observation: Depending on whether n is odd or even, we have

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n} = \left(\begin{array}{c|c|c} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n'} & \begin{matrix}\mathbf{A}\\ \mathbf{B}\end{matrix} & \\ \hline & & \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n'} \end{array}\right) \quad \text{or} \quad \left(\begin{array}{c|c|c} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n'} & \mathbf{A} & \\ \hline & \mathbf{B}\ \ \mathbf{A} & \\ \hline & \mathbf{B} & \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n'} \end{array}\right).$$

The protocol $\Pi_0^{\mathtt{fold}}.\langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}})\rangle$ consists of $\ell + 1$ rounds and makes use of the subtractive set $S \subset \mathcal{R}^\times$ mentioned in Section 4.3.1. Denote $(\mathbf{C}^{(0)}, \mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \mathbf{z}^{(0)}, \alpha^{(0)}) \coloneqq (\mathbf{C}, \mathbf{x}, \mathbf{y}, \mathbf{z}, \alpha)$. Express n uniquely as $n = \sum_{j=0}^{\ell} 2^j \cdot k_j$ where $k_j \in \{1, 2\}$. Note that $\mathbf{y}$ consists of $n' \coloneqq n + 1 = \sum_{j=0}^{\ell-1} 2^j \cdot (k_j - 1) + 2^\ell \cdot (k_\ell + 1)$ blocks. For $i \in \{0, \ldots, \ell\}$, define $n_i \coloneqq \sum_{j=i}^{\ell} 2^{j-i} \cdot k_j$ and $n_i' \coloneqq \sum_{j=i}^{\ell-1} 2^{j-i} \cdot (k_j - 1) + 2^{\ell-i} \cdot (k_\ell + 1)$. Then, for $i < \ell$, the i-th round of the protocol is as follows:

- Parse $(\mathbf{C}^{(i)}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ as

  $$(\mathbf{C}_L^{(i)}, \mathbf{C}_c^{(i)}, \mathbf{C}_R^{(i)}), \qquad (\mathbf{x}_L^{(i)}, \mathbf{x}_c^{(i)}, \mathbf{x}_R^{(i)}), \qquad \text{and} \qquad (\mathbf{y}_L^{(i)}, \mathbf{y}_c^{(i)}, \mathbf{y}_R^{(i)})$$

  respectively where $\mathsf{ncol}(\mathbf{C}_L^{(i)}) = \mathsf{ncol}(\mathbf{C}_R^{(i)}) = \mathsf{nrow}(\mathbf{x}_L^{(i)}) = \mathsf{nrow}(\mathbf{x}_R^{(i)}) = n_i \cdot w$ and $\mathsf{nrow}(\mathbf{y}_L^{(i)}) = \mathsf{nrow}(\mathbf{y}_R^{(i)}) = n_i' \cdot h$. Note that $\mathsf{nrow}(\mathbf{x}_c^{(i)}) = k_i$ and $\mathsf{nrow}(\mathbf{y}_c^{(i)}) = k_i - 1$, meaning that $\mathbf{y}_c^{(i)}$ is empty when $k_i = 1$.
- $\mathcal{P}$ sends

  $$\mathbf{x}_c^{(i)}, \qquad \mathbf{z}_{LR}^{(i)} \coloneqq \mathbf{C}_L^{(i)} \cdot \mathbf{x}_R^{(i)} \bmod q_1, \qquad \text{and} \qquad \mathbf{z}_{RL}^{(i)} \coloneqq \mathbf{C}_R^{(i)} \cdot \mathbf{x}_L^{(i)} \bmod q_1.$$

- $\mathcal{V}$ checks that $\left\|\mathbf{x}_c^{(i)}\right\| \leq \alpha^{(i)}$. If $k_i = 2$, $\mathcal{V}$ further checks that $\begin{pmatrix}\mathbf{B} & \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} = \mathbf{y}_c^{(i)} \bmod q_0$. If any of these checks fails, $\mathcal{V}$ aborts.
- $\mathcal{V}$ samples $r_i \leftarrow S$ and sends r_i to $\mathcal{P}$.

- $\mathcal{P}$ computes the compressed witness $\mathbf{x}^{(i+1)} \coloneqq \mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i$.
- $\mathcal{P}$ and $\mathcal{V}$ compute the compressed statement

$$\begin{aligned}
\mathbf{C}^{(i+1)} &\coloneqq \mathbf{C}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot r_i^{-1} \bmod q_1 \\
\mathbf{y}^{(i+1)} &\coloneqq \mathbf{y}_L^{(i)} + \mathbf{y}_R^{(i)} \cdot r_i - \begin{pmatrix} \mathbf{B} \cdot r_i \\ \mathbf{0} \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_c^{(i)} \bmod q_0 \\
\mathbf{z}^{(i+1)} &\coloneqq \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{z}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{z}_{LR}^{(i)} \cdot r_i \bmod q_1 \\
\alpha^{(i+1)} &\coloneqq 2 \cdot \alpha^{(i)} \cdot \gamma_{\mathcal{R}}
\end{aligned}$$

In the ℓ-th (i.e. final) round, $\mathcal{P}$ sends $\mathbf{x}^{(\ell)}$ and $\mathcal{V}$ checks that

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow k_\ell} \cdot \mathbf{x}^{(\ell)} = \mathbf{y}^{(\ell)} \bmod q_0, \qquad \text{and} \qquad \left\| \mathbf{x}^{(\ell)} \right\| \leq \alpha^{(\ell)} = (2\gamma_{\mathcal{R}})^{\ell} \cdot \alpha.$$
$$\mathbf{C}^{(\ell)} \cdot \mathbf{x}^{(\ell)} = \mathbf{z}^{(\ell)} \bmod q_1,$$

Analysis. We show that $\Pi_0^{\mathtt{fold}}$ is complete and (unconditionally) special-sound. We further show that $\Pi_0^{\mathtt{fold}}$ has short proofs, quasi-linear-time prover, and polylogarithmic-time verifier. The proofs of the above claims are deferred to Appendix C.3.

Theorem 4.7.1. *$\Pi_0^{\mathtt{fold}}$ is complete for $\Psi_0^{\mathtt{fold}}[\alpha]$.*

Theorem 4.7.2. *For $\alpha^* \geq (8\gamma_{\mathcal{R}}^4)^{\log n} \cdot \alpha$, $\Pi_0^{\mathtt{fold}}$ is $(3, \ldots, 3)$-special sound for $\Psi_0^{\mathtt{fold}}[\alpha^*]$.*

For the purpose of estimating the complexities of $\Pi_0^{\mathtt{fold}}$, let $h_0, h_1, w, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ be fixed polynomials in λ. Pick α^* to be tight in Theorem 4.7.2 and set $q_0, q_1 = O_\lambda(\alpha^*) = \lambda^{O(\log n)}$. The following theorem states the complexities of $\Pi_0^{\mathtt{fold}}$ with the above parameter choices.

Theorem 4.7.3. *Let $h_0, h_1, w, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ be fixed polynomials in λ, and $q_0, q_1 = \lambda^{O(\log n)}$. $\Pi_0^{\mathtt{fold}}$ has 1. prover time $O_\lambda(n \cdot \log^2 n)$, and 2. proof size $O_\lambda(\log^2 n)$. If $\mathbf{C}$ is $(k_0, \ldots, k_\ell)$-block-foldable with block-size w and $\mathbf{y}$ is $(k_0 - 1, \ldots, k_{\ell-1} - 1, k_\ell + 1)$-block-foldable with block-size h_0, then the verifier time is $O_\lambda(\log^3 n)$.*

4.7.2 Type-1 Linear Relations

Define the relation $\Psi_1^{\mathtt{fold}} = \Psi_1^{\mathtt{fold}}[\mathcal{R}, h, w, n, q, \alpha]$:

$$\Psi_1^{\mathtt{fold}} \coloneqq \left\{ (\mathsf{pp}, (\mathbf{A}, \mathbf{y}), \mathbf{x}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q \quad \text{and} \quad \|\mathbf{x}\| \leq \alpha \right\}$$

where $\mathcal{R}$ is a prime-power ring for a prime ≥ 5, $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_n)$, $\mathbf{A}_i \in \mathcal{R}_q^{h \times w}$, $\mathbf{y} \in \mathcal{R}_q^h$, and $\mathbf{x} \in \mathcal{R}^{wn}$. By default, we suppress all parameters of $\Psi_1^{\mathtt{fold}}$ except those that we highlight. Note that the above constraints are independent of pp, therefore $\Psi_1^{\mathtt{fold}}$ is compatible with any parameter generator Gen. We describe a protocol $\Pi_1^{\mathtt{fold}}$ which is complete for $\Psi_1^{\mathtt{fold}}[\alpha]$ and knowledge sound for $\Psi_1^{\mathtt{fold}}[\alpha^*]$ for some $\alpha^* > \alpha$.

Construction. We construct in Appendix C.2 a protocol $\Pi_1^{\mathtt{fold}}$ which can be seen as a simplification of $\Pi_0^{\mathtt{fold}}$ by removing components responsible for the structured part of the relation.

Analysis. We state the formal claims about the completeness, special-soundness, and efficiency of $\Pi_1^{\mathtt{fold}}$. The proofs of these claims are almost identical to those of Theorems 4.7.1 to 4.7.3 and are therefore omitted.

Theorem 4.7.4. *$\Pi_1^{\mathtt{fold}}$ is complete for $\Psi_1^{\mathtt{fold}}[\alpha]$.*

Theorem 4.7.5. *For $\alpha^* \geq (8\gamma_{\mathcal{R}}^4)^{\log n} \cdot \alpha$, $\Pi_1^{\mathtt{fold}}$ is $(3,\ldots,3)$-special sound for $\Psi_1^{\mathtt{fold}}[\alpha^*]$.*

Theorem 4.7.6. *Let $h, w = \mathsf{poly}(\lambda)$ and $q = \lambda^{O(\log n)}$. $\Pi_1^{\mathtt{fold}}$ has 1. prover time $O_\lambda(n \cdot \log^2 n)$, and 2. proof size $O_\lambda(\log^2 n)$. If $\mathbf{A}$ is $(k_0,\ldots,k_\ell)$-block-foldable with block-size w, then the verifier time is $O_\lambda(\log^3 n)$.*

4.8 Knowledge-based Protocols

Mirroring the folding protocols constructed in Section 4.7, we present below two argument systems $\Pi_0^{\mathtt{know}}$ and $\Pi_1^{\mathtt{know}}$ for unstructured linear relations based the (knowledge-)k-R-ISIS assumptions. Different from existing protocols based on the same family of assumptions and construction template, the constructions below feature quasi-linear-time provers.

4.8.1 Linear Relations

Define the relation $\Psi_0 = \Psi_0[\mathcal{R}, s, t, q_0, q_1, q_3, \alpha]$:

$$\Psi_0 := \left\{ ((\mathbf{v},\mathbf{h}),((\mathbf{M},\mathbf{y}),(c_{\mathbf{x}},\bar{c}_{\mathbf{x}})),\mathbf{x}) : \begin{array}{r} \mathbf{M}\cdot\mathbf{x} = \mathbf{y} \bmod q_0, \\ \mathbf{v}^{\mathsf{T}}\cdot\mathbf{x} = c_{\mathbf{x}} \bmod q_3, \\ (\bar{\mathbf{v}}\circ\mathbf{h})^{\mathsf{T}}\cdot\mathbf{x} = \bar{c}_{\mathbf{x}} \bmod q_3, \end{array} \quad \|\mathbf{x}\| \leq \alpha \right\}$$

where $\mathbf{M} \in \mathcal{R}_{q_3}^{t\times s}$, $\mathbf{y} \in \mathcal{R}_{q_3}^t$, $c_{\mathbf{x}}, \bar{c}_{\mathbf{x}} \in \mathcal{R}_{q_3}$, $\mathbf{x} \in \mathcal{R}^s$, $\mathbf{v} = (v, v^2, \ldots, v^s)$, and $\bar{\mathbf{v}} = (v^{-1}, v^{-2}, \ldots, v^{-s})$. Accompanying the relation, we define a parameter generator $\mathsf{Gen}^{\mathtt{unstr}}$ which samples $v \leftarrow \mathcal{R}_{q_3}^{\times}$ and $\mathbf{h} \leftarrow \mathcal{R}_{q_1}^s$ and outputs $(\mathbf{v},\mathbf{h})$. Note that the compression vector $\mathbf{h}$ is unstructured. By default, we suppress all parameters of Ψ_0 except those that we highlight. We describe a protocol $\Pi_0^{\mathtt{know}}$ which is complete for $\Psi_0[\alpha]$ and knowledge sound for $\Psi_0[\alpha^*]$ for some $\alpha^* > \alpha$.

Construction. Let $\mathcal{R}, s, t, \eta, m, (q_i)_{i=0}^3, \beta, (\delta_i)_{i=0}^3, \mathcal{T}$ depend on λ. Using the lattice trapdoor algorithms (section 4.3.2) parametrised by (η, m, q_3, β), in fig. 4.1 we give a formal description of $\Pi_0^{\mathtt{know}}$, which is based on the construction template of functional commitments in [ACL+22]. In particular, in $\Pi_0^{\mathtt{know}}$ the prover proves to the verifier that they know witnesses to the following relations

$$\begin{pmatrix} \mathbf{v}^{\mathsf{T}} \\ (\bar{\mathbf{v}}\circ\mathbf{h})^{\mathsf{T}} \end{pmatrix} \cdot \mathbf{x} = \begin{pmatrix} c_{\mathbf{x}} \\ \bar{c}_{\mathbf{x}} \end{pmatrix} \bmod q_3, \quad \text{and} \quad \|\mathbf{x}\| \leq \alpha, \tag{4.3}$$

$$\mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{r} = c_{\mathbf{r}}, \quad \text{with} \quad \mathbf{r} \in \mathcal{R}^t, \tag{4.4}$$

and

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \quad \|\mathbf{x}\| \leq \alpha. \tag{4.5}$$

The prover will prove that $c_{\mathbf{x}}$, $\bar{c}_{\mathbf{x}}$, and $c_{\mathbf{r}}$ are well-formed by proving knowledge of a short opening of the commitments $c_{\mathbf{x}}$, $\bar{c}_{\mathbf{x}}$, and $c_{\mathbf{r}}$ with respect to the commitment key $(v^i)_{i\in[s]}$, $(v^{-i})_{i\in[s]}$, and $(v_i)_{i\in[t]}$ respectively. To prove consistency between $c_{\mathbf{x}}$ and $\bar{c}_{\mathbf{x}}$, the prover proves knowledge of a short opening of the commitment $\bar{c}_{\mathbf{I}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{x}} \cdot c_{\mathbf{I}}$, where the values $\bar{c}_{\mathbf{I}}$ and $c_{\mathbf{I}}$ can be precomputed by the verifier. This is with respect to the commitment key $(v^k)_{k\in\pm[\max\{s,t\}]}$. Finally, to prove eq. (4.5), the prover proves knowledge of a short opening of the commitment $\bar{c}_{\mathbf{M}} \cdot c_{\mathbf{x}} + \bar{c}_{q_0} \cdot c_{\mathbf{r}} - \hat{c}_{\mathbf{y}}$, where the values $\bar{c}_{\mathbf{M}}, \bar{c}_{q_0}$, and $\hat{c}_{\mathbf{y}}$ can be precomputed by the verifier. This is again with respect to the commitment key $(v^k)_{k\in\pm[\max\{s,t\}]}$.

We highlight a few crucial differences with [ACL+22]:

1. The witness $\mathbf{x}$ is committed using a univariate vSIS commitment, i.e. the commitment key is $\mathbf{v} = (v, v^2, \ldots, v^s)$, while in [ACL+22] the commitment is an s-variate vSIS commitment. The fact that $|\{ v^{i-j} : i, j \in [s] \}|$ has cardinality $O(s)$ and that the prover computation consists of mainly Toeplitz-vector multiplications are crucial for obtaining a quasi-linear-time prover.

2. We support proving relations modulo q_0 natively[16] by introducing the auxiliary witness $\mathbf{r}$ satisfying $\mathbf{M} \cdot \mathbf{x} + q_0 \cdot \mathbf{r} = \mathbf{y}$. In [ACL+22], modular arithmetic is handled via generic and expensive bit-decomposition techniques.

3. To prove that values committed in multiple commitments, i.e. $c_{\mathbf{x}}$, $\bar{c}_{\mathbf{x}}$, and $c_{\mathbf{r}}$, satisfy some relation, we adapt techniques developed for the recent construction of chainable functional commitments [BCFL22].

Analysis. We show that $\Pi_0^{\mathtt{know}}$ is correct and knowledge-sound under (knowledge-)k-R-ISIS and R-SIS assumptions. We further show that $\Pi_0^{\mathtt{know}}$ has short CRS and proofs, quasi-linear-time prover and preprocessing, and polylogarithmic-time verifier after preprocessing. The proofs are deferred to Appendix C.5.

Theorem 4.8.1 (Completeness)**.** *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 4.3.2 hold. For*

$$\delta_0 \geq (s+t)^4 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3, \qquad \delta_1 \geq s \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}},$$
$$\delta_2 \geq s \cdot q_1 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}, \qquad \textit{and} \qquad \delta_3 \geq s^2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2,$$

$\Pi_0^{\mathtt{know}}$ in Figure 4.1 is complete for $\Psi_0[\alpha]$.

[16] Relations without modular reduction are captured by setting $q_0 = 0$.

$\mathsf{Setup}(1^\lambda, \mathsf{pp})$

$(\mathbf{v}, \mathbf{h}) \leftarrow \mathsf{pp}$
$\mathbf{f}_0 \leftarrow \mathcal{R}^t_{q_2},\ \mathbf{f}_1 \leftarrow \mathcal{R}^s_{q_2}$
$I_0 := \pm[\max\{s,t\}],\ I_1 := [s],\ I_2 := -[s],\ I_3 := [t]$
for $i \in \{0,1,2,3\}$ **do**
  $(\mathbf{D}_i, \mathsf{td}_i) \leftarrow \mathsf{TrapGen}(1^\lambda)$
  $\mathbf{t}_i \leftarrow \mathcal{T}$
  $\mathbf{u}_{i,j} \leftarrow \mathsf{SampPre}(\mathsf{td}_i, \mathbf{t}_i \cdot v^j),\ \forall j \in I_i$
$\mathsf{crs} := \begin{pmatrix} (\mathbf{D}_i, & \mathbf{t}_i, & (\mathbf{u}_{i,j})_{j\in I_i})_{i=0}^3, \\ v & \mathbf{h} & \mathbf{f}_0, \quad \mathbf{f}_1 \end{pmatrix}$
return crs

$\mathsf{Prove}(\mathsf{crs}, ((\mathbf{M}, \mathbf{y}), (c_\mathbf{x}, \bar{c}_\mathbf{x})), \mathbf{x})$

$\mathbf{v}_t := (v, v^2, \ldots, v^t)$
$c_\mathbf{r} := \mathbf{v}_t^\mathsf{T} \cdot \mathbf{r} \bmod q_3$
$\hat{\mathbf{u}}_{0,j} = \sum_{k\in[n], j\neq k} \mathbf{u}_{0,k-j} \cdot x_k,\ \forall j \in [n]$
$\mathbf{u}_\mathbf{M} := \sum_{i\in[t], j\in[s]} f_i \cdot M_{i,j} \cdot \hat{\mathbf{u}}_{0,j}$
$\mathbf{u}_\mathbf{M} := \sum_{i\in[t], j\in[s]} f_i \cdot M_{i,j} \cdot \sum_{k\in[s], k\neq j} \mathbf{u}_{0,k-j} \cdot x_k$
$\bar{\mathbf{u}} := \sum_{i,j\in[s], i\neq j} \mathbf{u}_{0,i-j} \cdot h_j \cdot (l_j \cdot x_i - l_i \cdot x_j)$
$\mathbf{u}_{0,0} := \sum_{i\in[s], k\in[t]} f_{0,k} \cdot M_{k,i} \cdot \sum_{j\in[s]:j\neq i} \mathbf{u}_{0,j-i} \cdot x_j$
$\quad + \sum_{i,k\in[t]} f_{0,k} \cdot q_0 \cdot \sum_{j\in[t]:j\neq i} \mathbf{u}_{0,j-i} \cdot r_j$
$\mathbf{u}_{0,1} := \sum_{i,j\in[s]:i\neq j} \mathbf{u}_{0,i-j} \cdot h_j \cdot (f_{1,j} \cdot x_i - f_{1,i} \cdot x_j)$
$\mathbf{u}_{0,1} := \sum_{j\in[s]} h_j \cdot f_{1,j} \cdot \sum_{i\in[s]:i\neq j} \mathbf{u}_{0,i-j} \cdot x_i$
$\quad - \sum_{i\in[s]} f_{1,i} \cdot \sum_{j\in[s]:j\neq i} \mathbf{u}_{0,i-j} \cdot h_j \cdot x_j$
$\mathbf{u}_{0,1} := \sum_{i,j\in[s]:i\neq j} l_i(h_i \cdot \mathbf{u}_{0,j-i} - v^{i-j} \cdot h_j) \cdot x_j$
$\mathbf{u}_0 := \mathbf{u}_{0,0} + \mathbf{u}_{0,1}$
$\mathbf{u}_1 := \sum_{j\in[s]} \mathbf{u}_{1,j} \cdot x_j$
$\mathbf{u}_2 := \sum_{j\in[s]} \mathbf{u}_{2,-j} \cdot h_j \cdot x_j$
$\mathbf{u}_3 := \sum_{j\in[t]} \mathbf{u}_{3,j} \cdot r_j$
return $\pi := (c_\mathbf{x}, \bar{c}_\mathbf{x}, c_\mathbf{r}, \mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$

$\mathsf{PreVerify}(\mathsf{crs}, (\mathbf{M}, \mathbf{y}))$

$\mathbf{v} := (v, v^2, \ldots, v^s),\ \bar{\mathbf{v}} := (v^{-1}, v^{-2}, \ldots, v^{-s})$
$\bar{\mathbf{v}}_t := (v^{-1}, v^{-2}, \ldots, v^{-t})$
$\bar{c}_\mathbf{M} := \mathbf{f}_0^\mathsf{T} \cdot \mathbf{M} \cdot \bar{\mathbf{v}} \bmod q_3$
$\bar{c}_{q_0} := \mathbf{f}_0^\mathsf{T} \cdot q_0 \cdot \bar{\mathbf{v}}_t \bmod q_3$
$\bar{c}_\mathbf{I} := \mathbf{f}_1^\mathsf{T} \cdot \mathbf{I} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) = \mathbf{f}_1^\mathsf{T} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \bmod q_3$
$c_\mathbf{I} := \mathbf{v}^\mathsf{T} \cdot \mathbf{I} \cdot \mathbf{f}_1 = \mathbf{v}^\mathsf{T} \cdot \mathbf{f}_1 \bmod q_3$
$\hat{c}_\mathbf{y} := \mathbf{f}_0^\mathsf{T} \cdot \mathbf{y} \bmod q_3$
$\mathsf{pp}_{\mathbf{M},\mathbf{y},c_\mathbf{x},\bar{c}_\mathbf{x}} := ((\mathbf{D}_i, \mathbf{t}_i)_{i=0}^3, \bar{c}_\mathbf{M}, \bar{c}_{q_0}, \bar{c}_\mathbf{I}, c_\mathbf{I}, \hat{c}_\mathbf{y})$
return $\mathsf{pp}_{\mathbf{M},\mathbf{y}}$

$\mathsf{Verify}(\mathsf{crs}_{\mathbf{M},\mathbf{y}}, (c_\mathbf{x}, \bar{c}_\mathbf{x}), \pi)$

$c_{0,0} := \bar{c}_\mathbf{M} \cdot c_\mathbf{x} + \bar{c}_{q_0} \cdot c_\mathbf{r} - \hat{c}_\mathbf{y} \bmod q_3$
$c_{0,1} := \bar{c}_\mathbf{I} \cdot c_\mathbf{x} - \bar{c}_\mathbf{x} \cdot c_\mathbf{I} \bmod q_3$
$c_0 := c_{0,0} + c_{0,1} \bmod q_3$
$c_1 := c_\mathbf{x}$
$c_2 := \bar{c}_\mathbf{x}$
$c_3 := c_\mathbf{r}$
for $i \in \{0,1,2,3\}$ **do**
  $b_i := (\mathbf{D}_i \cdot \mathbf{u}_i \stackrel{?}{\equiv} \mathbf{t}_i \cdot c_i \bmod q_3 \ \wedge\ \|\mathbf{u}_i\| \leq \delta_i)$
return $b_0 \wedge b_1 \wedge b_2 \wedge b_3$

Figure 4.1: Our argument system $\Pi_0^{\mathtt{know}}$.

Theorem 4.8.2 (Knowledge Soundness). *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 4.3.2 hold. Let $w = 1$, $\mathcal{G}_0 = \{ X^i : i \in \pm[\max\{s,t\}] \}$, $\mathcal{G}_1 = \{ X^i : i \in [s] \}$, $\mathcal{G}_2 = \{ X^i : i \in -[s] \}$, and $\mathcal{G}_3 = \{ X^i : i \in [t] \}$ be sets of monomials in X. Let $\mathcal{D}$ denote the distribution $\mathsf{SampD}(1^\lambda)$. For $i \in \{1,2,3\}$, let $\mathcal{Z}_i(1^\lambda)$ be almost identical to $\mathsf{Setup}(1^\lambda, \mathsf{Gen}^{\mathtt{unstr}}(1^\lambda))$, except that it inputs $(\mathbf{D}_i, \mathbf{t}_i, v, \{\mathbf{u}_{i,j}\}_{j \in I_i})$ and generates the rest of crs. Let*

$$\alpha_i^* \geq \delta_i, \ \forall i \in [3], \qquad \alpha^* := \max\{\alpha_1^*, \alpha_2^*, \alpha_3^*\}, \qquad q_2 \geq \beta_{q_2}^* \geq s \cdot q_0 \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}},$$
$$q_3 \geq \beta_{q_3}^* \geq \max\{2\delta_0, (s+t)^3 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \beta \cdot \gamma_{\mathcal{R}}^3\}.$$

$\Pi_0^{\mathtt{know}}$ in Figure 4.1 is knowledge-sound for $\Psi_0[\alpha_1^*]$ if the following assumptions hold:

Assumption 0. *k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\beta,\beta_{q_3}^*,\mathcal{G}_0,g^*=1,\mathcal{D},\mathcal{T}}$,*

Assumption 1. *knowledge-k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_1^*,\beta,\delta_1,\mathcal{G}_1,\mathcal{D},\mathcal{T},\mathcal{Z}_1}$,*

Assumption 2. *knowledge-k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_2^*,\beta,\delta_2,\mathcal{G}_2,\mathcal{D},\mathcal{T},\mathcal{Z}_2}$,*

Assumption 3. *knowledge-k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_3^*,\beta,\delta_3,\mathcal{G}_3,\mathcal{D},\mathcal{T},\mathcal{Z}_3}$, and*

Assumption 4. *R-$\mathsf{SIS}_{\mathcal{R},s+t,q_2,\beta_{q_2}^*}$.*

For the purpose of estimating complexities of the scheme, we assume that the assumptions in Theorem 4.8.2 hold for moduli which are a fixed polynomial factor larger than their norm bounds, e.g. $q_2 \geq \beta_{q_2}^* \cdot \mathsf{poly}(\lambda)$ for the R-$\mathsf{SIS}_{\mathcal{R},s+t,q_2,\beta_{q_2}^*}$ assumption. For the k-R-ISIS assumptions, we assume that they hold for $m = O(\eta \cdot \log q)$.

Let $\eta, \alpha, \beta, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ be fixed polynomials in λ. For our application in Section 4.9, we want $q_1 = O(s^2 \cdot \alpha^2) = O_\lambda(s^2)$. Pick $\delta_1, \delta_2, \delta_3, \alpha_1^*, \alpha_2^*, \alpha_3^*$ so that they match their lower bounds given in Theorem 4.8.1 and Theorem 4.8.2 respectively. Substituting q_1, we have $\alpha_1^* = \delta_1 = O_\lambda(s)$, $\alpha_2^* = \delta_2 = O_\lambda(s^3)$, and $\alpha_3^* = \delta_3 = O_\lambda(s^2)$. We therefore have $\alpha^* = O_\lambda(s^3)$. Pick $q_0 = O_\lambda(\alpha_1^*) = O_\lambda(s)$. Pick $\beta_{q_2}^*$ so that it matches its lower bound in Theorem 4.8.2, and set $q_2 = O_\lambda(\beta_{q_2}^*)$. Substituting (q_0, q_1, α^*), we have $q_2 = O_\lambda(s^7)$. Pick δ_0 so that it matches its lower bound given in Theorem 4.8.1. Substituting (q_0, q_1, q_2), we have $\delta_0 = O_\lambda((s+t)^{14})$. Pick $\beta_{q_3}^*$ so that it matches its lower bound in Theorem 4.8.2, and set $q_3 = O_\lambda(\beta_{q_3}^*)$. Substituting $(q_0, q_1, q_2, \alpha^*)$, we have $q_3 = O_\lambda((s+t)^{16})$. Let $n = \max\{|\mathbf{M}|, s+t\}$, where $|\mathbf{M}|$ denote the number of non-zero entries in $\mathbf{M}$. Pick $m = O(\eta \cdot \log q) = O_\lambda(\log n)$.

The following theorem states the complexities of the scheme with the above parameter choices.

Theorem 4.8.3 (Efficiency). *Let $n = \max\{|\mathbf{M}|, s+t\}$, where $|\mathbf{M}|$ denote the number of non-zero entries in $\mathbf{M}$, $\eta, \alpha, \beta, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ be fixed polynomials in λ, and $(m, q_0, q_1, q_2, q_3) = (\log n, s, s^2, s^7, (s+t)^{16}) \cdot \mathsf{poly}(\lambda)$. Then $\Pi_0^{\mathtt{fold}}$ has 1. common reference string size $O_\lambda(n \cdot \log n)$, 2. proof size $O_\lambda(\log^2 n)$, 3. prover time $O_\lambda(n \cdot \log^3 n)$, 4. preprocessing time $O_\lambda(n \cdot \log^2 n)$, and 5. verifier time $O_\lambda(\log^3 n)$ after preprocessing.*

4.8.2 Well-formedness of vSIS Commitments

Define the relation $\Psi_1 = \Psi_1[\mathcal{R}, s, q_1, q_3, \alpha]$ equipped with the same parameter generator $\mathsf{Gen}^{\mathtt{unstr}}$ as Ψ_0:

$$\Psi_1 := \left\{((\mathbf{v}, \mathbf{h}), (\epsilon, c_{\mathbf{z}}), \mathbf{z}) : \begin{pmatrix} \bar{\mathbf{v}}^{\mathsf{T}} & \mathbf{v}^{\mathsf{T}} \end{pmatrix} \cdot \mathbf{z} = c_{\mathbf{z}} \bmod q_3 \quad \wedge \quad \|\mathbf{z}\| \leq \alpha \right\}$$

where $c_{\mathbf{z}} \in \mathcal{R}_{q_3}$, $\mathbf{z} \in \mathcal{R}^{2s}$, $\mathbf{v} = (v, v^2, \ldots, v^s)$, and $\bar{\mathbf{v}} = (v^{-1}, v^{-2}, \ldots, v^{-s})$. By default, we suppress all parameters of Ψ_1 except those that we highlight. We describe a protocol $\Pi_1^{\mathtt{know}}$ which is complete for $\Psi_1[\alpha]$ and knowledge sound for $\Psi_1[\alpha^*]$ for some $\alpha^* > \alpha$.

Construction. We construct in Appendix C.4 a protocol $\Pi_1^{\mathtt{know}}$ for the relation Ψ_1. The proof for $\mathbf{z}$ is simply $\mathbf{D}^{-1}(\mathbf{t} \cdot c_{\mathbf{z}})$ with $(\mathbf{D}, \mathbf{t})$ given in crs.

Analysis. $\Pi_1^{\mathtt{know}}$ is correct and knowledge-sound under the knowledge-k-R-ISIS assumption. It has short CRS and proofs, quasi-linear-time prover, and polylogarithmic-time verifier. Below, we state these claims formally and omit the (trivial) proofs.

Theorem 4.8.4 (Completeness)**.** *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 4.3.2 hold. For $\delta \geq 2s \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}$ $\Pi_1^{\mathtt{know}}$ in Figure 4.1 is complete for $\Psi_1[\alpha]$.*

Theorem 4.8.5 (Knowledge Soundness)**.** *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 4.3.2 hold, $w = 1$, $\alpha^* \geq \delta$, $\mathcal{G} = \{ X^i : i \in \pm[s] \}$ be a set of monomials in X, $\mathcal{D}$ denote the distribution $\mathsf{SampD}(1^\lambda)$, and $\mathcal{Z}$ be trivial (i.e. it outputs $\perp$). $\Pi_1^{\mathtt{know}}$ in Figure 4.1 is knowledge-sound for $\Psi_0[\alpha^*]$ if the knowledge-k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha^*,\beta,\delta,\mathcal{G},\mathcal{D},\mathcal{T},\mathcal{Z}}$ assumption holds.*

Theorem 4.8.6 (Efficiency)**.** *Let parameters be as in Theorem 4.8.3. $\Pi_1^{\mathtt{fold}}$ has 1. common reference string size $O_\lambda(n \cdot \log n)$, 2. proof size $O_\lambda(\log^2 n)$, 3. prover time $O_\lambda(n \cdot \log^2 n)$, 4. trivial preprocessing, and 5. verifier time $O_\lambda(\log^3 n)$.*

4.9 Applications

We show how to compose arguments obtain in Sections 4.7 and 4.8 to build efficient arguments for more complex relations. In particular, we show how to construct arguments for the binary-satisfiability of (structured) linear equations and rank-1 constraint satisfiability (R1CS).

4.9.1 Proving Binary-Satisfiability of (Structured) Linear Equations

Recall that in Section 4.7 we built succinct arguments $\Pi_0^{\mathtt{fold}}$ and $\Pi_1^{\mathtt{fold}}$ for the relations $\Psi_0^{\mathtt{fold}}$ and $\Psi_1^{\mathtt{fold}}$ respectively, while in Section 4.8 we constructed $\Pi_0^{\mathtt{know}}$ and $\Pi_1^{\mathtt{know}}$ for the relations Ψ_0 and Ψ_1 respectively. By inspection, we see that Ψ_1 is a special case of $\Psi_1^{\mathtt{fold}}$, and thus $\Pi_1^{\mathtt{fold}}$ can be specialised to give a succinct argument for Ψ_1. Similarly,

$\mathsf{Setup}(1^\lambda)$	$\mathsf{PreVerify}(\mathsf{crs}, (\mathbf{M}, \mathbf{y}))$
$\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda)$	$\mathsf{crs}'_{(\mathbf{M},\mathbf{y})} \leftarrow \Pi'.\mathsf{PreVerify}(\mathsf{crs}', (\mathbf{M}, \mathbf{y}))$
$\mathsf{crs}' \leftarrow \Pi'.\mathsf{Setup}(1^\lambda, \mathsf{pp})$	$\mathsf{crs}''_\epsilon \leftarrow \Pi''.\mathsf{PreVerify}(\mathsf{crs}'', \epsilon)$
$\mathsf{crs}'' \leftarrow \Pi''.\mathsf{Setup}(1^\lambda, \mathsf{pp})$	**return** $\mathsf{crs}_{(\mathbf{M},\mathbf{y})} := (\mathsf{crs}'_{(\mathbf{M},\mathbf{y})}, \mathsf{crs}''_\epsilon)$
return $\mathsf{crs} := (\mathsf{crs}', \mathsf{crs}'')$	

Figure 4.2: Setup and PreVerify algorithms of the argument system $\Pi^{\mathtt{bin\text{-}sat}}$.

$\Pi_0^{\mathtt{fold}}$ can be specialised as to give a succinct argument for the following special case of Ψ_0 which we denote by $\Psi_0^{\mathtt{str}} = \Psi_0^{\mathtt{str}}[\mathcal{R}, h, w, n, q_0, q_1, q_3, \alpha]$, where $\mathbf{M}$ is restricted to be of the form $\mathbf{M} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n}$ succinctly represented by some $\mathbf{A}, \mathbf{B} \in \mathcal{R}_{q_0}^{h\times w}$.

Accompanying $\Psi_0^{\mathtt{str}}$, we define the parameter generator $\mathsf{Gen}^{\mathtt{str}}$ which samples $(\mathbf{v}, \mathbf{h})$ which are $(k_0, \ldots, k_\ell)$-block-foldable with block-size w where $n = \sum_{i=0}^{\ell} k_i$ for $k_i \in \{1, 2\}$. More concretely, $\mathsf{Gen}^{\mathtt{str}}$ does the following: 1. Sample $v \leftarrow \mathcal{R}_q^\times$ and $\tilde{\mathbf{h}} \leftarrow \mathcal{R}_{q_1}^{\tilde{n}}$. 2. Set $\mathbf{v} := (v, \ldots, v^s) \bmod q_3$. 3. Let $\tilde{n} := \sum_{i=0}^{\ell-1}(k_i + 1) + k_\ell$. 4. Generate w copies of $\tilde{n}$-variate monomial sequences $\mathbf{m}_1, \ldots, \mathbf{m}_w$ according to Lemma 4.6.6, and concatenate them in an interleaved manner into a monomial sequence $\mathbf{m} = (m_{1,1}, m_{2,1}, \ldots, m_{w,1}, m_{1,2}, \ldots, m_{w,n})$. 5. Evaluate $\mathbf{m}$ at $\tilde{\mathbf{h}}$ to produce $\mathbf{h} = \mathbf{m}(\tilde{\mathbf{h}})$.

Equipped with succinct arguments for Ψ_0 (or $\Psi_0^{\mathtt{str}}$) and Ψ_1, we show how to construct a succinct argument $\Pi^{\mathtt{bin\text{-}sat}}$ for the binary-satisfiability of system of (structured) linear equations mod p. Formally, define the relation $\Psi^{\mathtt{bin\text{-}sat}} = \Psi^{\mathtt{bin\text{-}sat}}[\mathcal{R}, s, t, p]$:

$$\Psi^{\mathtt{bin\text{-}sat}} := \left\{(((\mathbf{M}, \mathbf{y}), \epsilon), \mathbf{x}) : \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \quad \wedge \quad \mathbf{x} \in \{0,1\}^s\right\},$$

where $\mathbf{M} \in \mathcal{R}_{q_0}^{t\times s}$, $\mathbf{y} \in \mathcal{R}_{q_0}^t$, and $\mathbf{x} \in \mathcal{R}^s$, and the corresponding structured variant $\Psi^{\mathtt{str\text{-}bin\text{-}sat}} = \Psi^{\mathtt{str\text{-}bin\text{-}sat}}[\mathcal{R}, h, w, n, p]$ where $\mathbf{M}$ is restricted to be of the form $\mathbf{M} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n}$ succinctly represented by some $\mathbf{A}, \mathbf{B} \in \mathcal{R}_{q_0}^{h\times w}$.

Let q_1, q_3 depend on λ. Let Π' and Π'' be argument systems for Ψ_0 (or $\Psi_0^{\mathtt{str}}$) and Ψ_1 respectively, and let $\mathsf{Gen} = \mathsf{Gen}^{\mathtt{unstr}}$ (or $\mathsf{Gen}^{\mathtt{str}}$) be the accompanying parameter generator. The algorithms $\Pi^{\mathtt{bin\text{-}sat}}.(\mathsf{Setup}, \mathsf{PreVerify})$ are described in Figure 4.2. The protocol $\Pi^{\mathtt{bin\text{-}sat}}.\left\langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{(\mathbf{M},\mathbf{y})}, \epsilon)\right\rangle$ is below:

- Prove computes
  1. $c_{\mathbf{x}} := \langle \mathbf{v}, \mathbf{x}\rangle \bmod q_3$,
  2. $\bar{c}_{\mathbf{x}} := \langle \bar{\mathbf{v}} \circ \mathbf{h}, \mathbf{x}\rangle \bmod q_3$, and

3. $\mathbf{z} \coloneqq \left(\sum_{0\le i,j\le s: i-j=k} h_j \cdot x_j \cdot (x_i - 1)\right)_{-s\le k\le s}$.

- Prove sends $(c_{\mathbf{x}}, \bar{c}_{\mathbf{x}})$ to Verify.
- Prove and Verify compute:
  - $c_{\mathbf{z}} \coloneqq \bar{c}_{\mathbf{x}} \cdot (c_{\mathbf{x}} - \langle \mathbf{v}, \mathbf{1}\rangle) \bmod q_3$.
  - $\mathsf{stmt}' \coloneqq ((\mathbf{M}, \mathbf{y}), (c_{\mathbf{x}}, \bar{c}_{\mathbf{x}}))$, $\mathsf{stmt}'' \coloneqq (\epsilon, c_{\mathbf{z}})$.
  - $(\mathsf{tx}', b') \leftarrow \Pi'. \left\langle \mathsf{Prove}(\mathsf{crs}', \mathsf{stmt}', \mathbf{x}), \mathsf{Verify}(\mathsf{crs}'_{(\mathbf{M},\mathbf{y})}, (c_{\mathbf{x}}, \bar{c}_{\mathbf{x}}))\right\rangle$.
  - $(\mathsf{tx}'', b'') \leftarrow \Pi''. \langle \mathsf{Prove}(\mathsf{crs}'', \mathsf{stmt}'', \mathbf{z}), \mathsf{Verify}(\mathsf{crs}''_{\epsilon}, c_{\mathbf{z}})\rangle$.
- Output (tx, b), where $\mathsf{tx} = (\mathsf{tx}', \mathsf{tx}'')$ and $b = b' \wedge b''$.

We show that $\Pi^{\texttt{bin-sat}}$ is complete and knowledge-sound. We further show that $\Pi^{\texttt{bin-sat}}$ has short proofs, quasi-linear-time prover, and polylogarithmic-time verifier (after preprocessing in the unstructured case). All proofs are deferred to Appendix C.6.

Theorem 4.9.1. *If* $\mathsf{Gen} = \mathsf{Gen}^{\texttt{str}}$ *(resp.* $\mathsf{Gen}^{\texttt{unstr}}$*),* Π' *is complete for* $\Psi_0^{\texttt{str}}[\alpha = 1]$*, and* Π'' *is complete for* $\Psi_1[\alpha = s \cdot (q_1/2)^{\ell+1} \cdot \gamma_{\mathcal{R}}^{\ell}]$ *(resp.* $\Psi_1[\alpha = s \cdot q_1/2]$*) then* $\Pi^{\texttt{bin-sat}}$ *is complete for* $\Psi^{\texttt{str-bin-sat}}$ *(resp.* $\Psi^{\texttt{-bin-sat}}$*).*

Theorem 4.9.2. *Let* $\mathsf{Gen} = \mathsf{Gen}^{\texttt{str}}$ *(resp.* $\mathsf{Gen}^{\texttt{unstr}}$*). Let* $\mathcal{G} \coloneqq \{ X^j : -s \le j \le s \}$ *and* $\mathcal{G}_{\mathbf{h}}$ *be the set of monomials generated as in* $\mathsf{Gen}^{\texttt{str}}$*. Let* $q_1, q_3, \alpha', \alpha'', \beta_{q_1}, \beta_{q_3}$ *be such that 1.* $\beta_{q_1} \ge (\alpha'+1)^2 \cdot \gamma_{\mathcal{R}}$*, 2.* $\beta_{q_3} \ge \alpha'' + s \cdot (q_1/2)^{\ell+1} \cdot (\alpha'+1)^2 \cdot \gamma_{\mathcal{R}}^{\ell+2}$ *(resp.* $\alpha'' + s \cdot q_1/2 \cdot (\alpha'+1)^2 \cdot \gamma_{\mathcal{R}}^2$*), 3.* Π' *is knowledge-sound for* $\Psi_0^{\texttt{str}}[\alpha']$ *(resp.* $\Psi_0^{\texttt{unstr}}[\alpha']$*), and 4.* Π'' *is knowledge-sound for* $\Psi_1[\alpha'']$*.* $\Pi^{\texttt{bin-sat}}$ *is knowledge-sound for* $\Psi^{\texttt{str-bin-sat}}$ *(resp.* $\Psi^{\texttt{bin-sat}}$*), if the following assumptions hold:*

Assumption 0. $\mathsf{vSIS}_{\mathcal{R},\mathcal{G}_{\mathbf{h}},1,q_1,\beta_{q_1}}$ *(resp.* $R\text{-}\mathsf{SIS}_{\mathcal{R},s,q_1,\beta_{q_1}}$*), and*

Assumption 1. $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},1,q_3,\beta_{q_3}}$.

Below, we estimate the complexities of $\Pi^{\texttt{bin-sat}}$ for parameters chosen in such a way that completeness and knowledge-soundness (are believed to) hold.

Theorem 4.9.3. *In the structured setting, let* $\mathsf{Gen} = \mathsf{Gen}^{\texttt{str}}$*,* $\Pi' = \Pi_0^{\texttt{fold}}$ *(specialised for* $\Psi_0^{\texttt{str}}$*),* $\Pi'' = \Pi_1^{\texttt{fold}}$ *(specialised for* Ψ_1*),* $\gamma_{\mathcal{R}}, \alpha', \alpha'', h, w = \mathsf{poly}(\lambda)$ *be fixed polynomials in* λ*, and* $q_0, q_1, q_3 = \lambda^{O(\log n)}$*.* $\Pi^{\texttt{bin-sat}}$ *has 1. common reference string size* $O_\lambda(\log^2 n)$*, 2. prover time* $O_\lambda(n \cdot \log^3 n)$*, and 3. proof size* $O_\lambda(\log^2 n)$*. If* $\mathbf{y}$ *is* $(k_0 - 1, \ldots, k_{\ell-1} - 1, k_\ell + 1)$*-block-foldable with block-size* h*, then the verifier time is* $O_\lambda(\log^3 n)$*.*

In the unstructured setting, let $\mathsf{Gen} = \mathsf{Gen}^{\texttt{unstr}}$*,* $\Pi' = \Pi_0^{\texttt{know}}$*,* $\Pi'' = \Pi_1^{\texttt{know}}$*,* $\gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ *be a fixed polynomial in* λ*,* $n = \max\{|\mathbf{M}|, s+t\}$ *where* $|\mathbf{M}|$ *denote the number of non-zero entries in* $\mathbf{M}$*,* $(q_0, q_1, q_3) = (s, s^2, (s+t)^{16}) \cdot \mathsf{poly}(\lambda)$ *and other internal parameters of* $\Pi_0^{\texttt{know}}$ *and* $\Pi_1^{\texttt{know}}$ *be chosen as in Theorems 4.8.3 and 4.8.6.* $\Pi^{\texttt{bin-sat}}$ *has 1. common reference string size* $O_\lambda(n \cdot \log n)$*, 2. proof size* $O_\lambda(\log^2 n)$*, 3. prover time* $O_\lambda(n \cdot \log^3 n)$*, 4. preprocessing time* $O_\lambda(n \cdot \log^2 n)$*, and 5. verifier time* $O_\lambda(\log^3 n)$ *after preprocessing.*

4.9.2 Rank-1 Constraint Systems

We show how to use the same ideas to construct an argument of knowledge, $\Pi_{\mathtt{R1CS}}$, for the satisfiability of Rank-1 Constraint Systems. Formally, define the relation $\Psi^{\mathtt{R1CS}} = \Psi^{\mathtt{R1CS}}[\mathcal{R}, t, s_1, s_2, q_0, \alpha]$:

$$\Psi^{\mathtt{R1CS}} \coloneqq \left\{((\mathbf{x}_1, \mathbf{E}, \mathbf{F}, \mathbf{G}), \mathbf{x}_2) : (\mathbf{E} \cdot \mathbf{x}) \circ (\mathbf{F} \cdot \mathbf{x}) = \mathbf{G} \cdot \mathbf{x} \bmod q_0 \quad \wedge \quad \|\mathbf{x}\| \leq \alpha \quad \right\},$$

where $\mathbf{x} \coloneqq (\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{R}^{s_1} \times \mathcal{R}^{s_2}$, $\mathbf{E}, \mathbf{F}, \mathbf{G} \in \mathcal{R}_{q_0}^{t \times s}$, and $s = s_1 + s_2$. If we let $\mathbf{e} \coloneqq \mathbf{E} \cdot \mathbf{x}$, $\mathbf{f} \coloneqq \mathbf{F} \cdot \mathbf{x}$, and $\mathbf{g} \coloneqq \mathbf{G} \cdot \mathbf{x}$, the above equation can be rewritten as

$$\mathbf{e} \circ \mathbf{f} + q_0 \cdot \mathbf{r} = \mathbf{g},$$

for some $\mathbf{r} \in \mathcal{R}^t$. For readability, we informally describe here how the argument system works. A formal description of $\Pi^{\mathtt{R1CS}}$ can be found in Figure C.2 in Appendix C.7.

In $\Pi_{\mathtt{R1CS}}$, the prover proves to the verifier that they know witnesses to the following relations

$$\mathbf{v}_2^{\mathsf{T}} \cdot \mathbf{x}_2 = c_{\mathbf{x}_2} \bmod q_3, \quad \text{and} \quad \|\mathbf{x}_2\| \leq \alpha, \tag{4.6}$$

where $\mathbf{v}_2 = (v^{s_1+1}, \ldots, v^s)$,

$$\begin{pmatrix} (\bar{\mathbf{v}}_t \circ \mathbf{h})^{\mathsf{T}} \cdot \mathbf{E} \\ \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{F} \\ \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{G} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \bar{c}_{\mathbf{e}} \\ c_{\mathbf{f}} \\ c_{\mathbf{g}} \end{pmatrix} \bmod q_3, \quad \text{and} \quad \left\| \begin{pmatrix} \mathbf{x}_2 \end{pmatrix} \right\| \leq \alpha, \tag{4.7}$$

where $\mathbf{h} \in \mathcal{R}_{q_1}^t$, and

$$\left(\bar{\mathbf{v}}^{\mathsf{T}} || \mathbf{v}^{\mathsf{T}}\right) \cdot \mathbf{z} = c_{\mathbf{z}} \bmod q_3, \quad \text{and} \quad \|\mathbf{z}\| \leq \alpha', \tag{4.8}$$

where $\mathbf{z} = (z_k)_{k \in \pm[s]}$, $z_k = \sum_{i,j,i-j=k} h_j \cdot e_j \cdot f_i + q_0 \cdot h_j \cdot r_i - g_i \cdot h_j$, $c_{\mathbf{z}} = \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot c_{\mathbf{r}} \cdot \bar{c}_{\mathbf{I}} - c_{\mathbf{g}} \cdot \bar{c}_{\mathbf{I}}$, and $c_{\mathbf{r}} = \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{r}$.

The prover will prove that $c_{\mathbf{x}_2}$ is well-formed, i.e, relation in Equation (4.6), by proving knowledge of a short opening of the commitment $c_{\mathbf{x}_2}$ with respect to the commitment key $(v_i)_{i \in [s_1+1;s]}$. To prove consistency between $c_{\mathbf{x}_2}$ and $\bar{c}_{\mathbf{e}}$, the prover proves knowledge of a short opening of the commitment

$$\bar{c}_{\mathbf{E}} \cdot c_{\mathbf{x}} - c_{\mathbf{I}} \cdot \bar{c}_{\mathbf{e}}$$

where $c_{\mathbf{x}} \coloneqq c_{\mathbf{x}_1} + c_{\mathbf{x}_2}$, and the values $c_{\mathbf{x}_1} \coloneqq \mathbf{v}_1^{\mathsf{T}} \cdot \mathbf{x}_1$, $\bar{c}_{\mathbf{E}}$, and $c_{\mathbf{I}}$ can be precomputed by the verifier. This with respect to the commitment key $(v^{i-j})_{i-j=k, k \in \pm[s]}$. Proofs of consistency between $c_{\mathbf{x}_2}$ and $c_{\mathbf{f}}$, $c_{\mathbf{x}_2}$ and $c_{\mathbf{g}}$ are obtained similarly. This suffices to prove the relation in Equation (4.7).

Finally, to prove that $\mathbf{e} \circ \mathbf{f} = \mathbf{g} \bmod q_0$, i.e., relation in Equation (4.8), the prover will prove knowledge of a short opening of the commitment

$$c_{\mathbf{z}} = \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot \bar{c}_{\mathbf{I}} \cdot c_{\mathbf{r}} - c_{\mathbf{g}} \cdot \bar{c}_{\mathbf{I}}$$

again with respect to the commitment key $(v^{i-j})_{i-j=k, k \in \pm[s]}$.

Analysis. In Appendix C.7 we show that $\Pi^{\mathtt{R1CS}}$ is complete and knowledge-sound under (knowledge-)k-R-ISIS and R-SIS assumptions. We further show that $\Pi^{\mathtt{R1CS}}$ has short CRS and proofs, quasi-linear-time prover and preprocessing, and polylogarithmic-time verifier after preprocessing. For readability, we defer formal claims and relative proofs to Appendix C.7.2, and Appendix C.7.3.

List of Figures

List of Tables

Bibliography

[AASA+20] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, et al. Status report on the second round of the nist post-quantum cryptography standardization process. *US Department of Commerce, NIST*, 2020.

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

[ABB+19] Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Phillipe Gaborit, Shay Gueron, Tim Guneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, Gilles Zémor, and Valentin Vasseur. BIKE. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions`.

[ABD+19] Nicolas Aragon, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Adrien Hauteville, Olivier Ruatta, Jean-Pierre Tillich, Gilles Zémor, Carlos Aguilar Melchor, Slim Bettaieb, Loic Bidoux, Magali Bardet, and Ayoub Otmani. ROLLO. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions`.

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed Σ-protocol theory for lattices. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part II*, volume 12826 of *Lecture Notes*

in Computer Science, pages 549–579, Virtual Event, August 16–20, 2021. Springer, Heidelberg, Germany.

[ACL+22] Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable - (extended abstract). In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 102–132, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Heidelberg, Germany.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *29th Annual ACM Symposium on Theory of Computing*, pages 284–293, El Paso, TX, USA, May 4–6, 1997. ACM Press.

[AEE+21] Lukas Aumayr, Oguzhan Ersoy, Andreas Erwig, Sebastian Faust, Kristina Hostáková, Matteo Maffei, Pedro Moreno-Sanchez, and Siavash Riahi. Generalized channels from limited blockchain scripts and adaptor signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 635–664. Springer, 2021.

[AF22] Thomas Attema and Serge Fehr. Parallel repetition of $(k_1, \ldots, k_\mu)$-special-sound multi-round interactive proofs. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 415–443, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Heidelberg, Germany.

[AGHS13] Shweta Agrawal, Craig Gentry, Shai Halevi, and Amit Sahai. Discrete Gaussian leftover hash lemma over infinite domains. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 97–116, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany.

[AGPS20] Martin R. Albrecht, Vlad Gheorghiu, Eamonn W. Postlethwaite, and John M. Schanck. Estimating quantum speedups for lattice sieves. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 583–613, Daejeon, South Korea, December 7–11, 2020. Springer, Heidelberg, Germany.

[Agr20] Shweta Agrawal. Unlikely friendships: The fruitful interplay of cryptography assumptions. Invited talk at ASIACRYPT 2020, December 2020. `https://youtu.be/Owz8UuWTsqg`.

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele

Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing*, pages 99–108, Philadephia, PA, USA, May 22–24, 1996. ACM Press.

[AKSY21] Shweta Agrawal, Elena Kirshanova, Damien Stehle, and Anshu Yadav. Can round-optimal lattice-based blind signatures be practical? Cryptology ePrint Archive, Report 2021/1565, 2021. `https://eprint.iacr.org/2021/1565`.

[AL21] Martin R. Albrecht and Russell W. F. Lai. Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 519–548, Virtual Event, August 16–20, 2021. Springer, Heidelberg, Germany.

[AME+21] Lukas Aumayr, Matteo Maffei, Oğuzhan Ersoy, Andreas Erwig, Sebastian Faust, Siavash Riahi, Kristina Hostáková, and Pedro Moreno-Sanchez. Bitcoin-compatible virtual channels. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 901–918. IEEE, 2021.

[APRS20] Daniel Apon, Ray A. Perlner, Angela Robinson, and Paolo Santini. Cryptanalysis of LEDAcrypt. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 389–418, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 430–454, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th Annual Symposium on Foundations of Computer Science*, pages 474–483, Philadelphia, PA, USA, October 18–21, 2014. IEEE Computer Society Press.

[ASS+16] Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, and Yuval Shavitt. DROWN: Breaking TLS using SSLv2. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX*

Security Symposium, pages 689–706, Austin, TX, USA, August 10–12, 2016. USENIX Association.

[Bab86] László Babai. On lovász' lattice reduction and the nearest lattice point problem. *Comb.*, 6(1):1–13, 1986.

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334, San Francisco, CA, USA, May 21–23, 2018. IEEE Computer Society Press.

[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 757–788, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

[BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[BBC+19] Marco Baldi, Alessandro Barenghi, Franco Chiaraluce, Gerardo Pelosi, and Paolo Santini. LEDAcrypt. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions`.

[BBC+20] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. Improvements of algebraic attacks for solving the rank decoding and minrank problems, 2020.

[BBD09] Daniel J Bernstein, Johannes Buchmann, and Erik Dahmen. Post-quantum cryptography.–2009, 2009.

[BBGP16] Johannes A Buchmann, Denis Butin, Florian Göpfert, and Albrecht Petzoldt. Post-quantum cryptography: state of the art. *The new codebreakers*, pages 88–108, 2016.

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 259–274, Bruges, Belgium, May 14–18, 2000. Springer, Heidelberg, Germany.

[BCC+09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 108–125, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Heidelberg, Germany.

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 111–120, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

[BCD+16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 1006–1018, Vienna, Austria, October 24–28, 2016. ACM Press.

[BCFL22] David Balbás, Dario Catalano, Dario Fiore, and Russell W. F. Lai. Functional commitments for circuits from falsifiable assumptions. Cryptology ePrint Archive, Report 2022/1365, 2022. `https://eprint.iacr.org/2022/1365`.

[BCG+13] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. SNARKs for C: Verifying program executions succinctly and in zero knowledge. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 90–108, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[BCG+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, Berkeley, CA, USA, May 18–21, 2014. IEEE Computer Society Press.

[BCG+19] Eli Ben-Sasson, Alessandro Chiesa, Lior Goldberg, Tom Gur, Michael Riabzev, and Nicholas Spooner. Linear-size constant-query IOPs for delegating computation. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 494–521, Nuremberg, Germany, December 1–5, 2019. Springer, Heidelberg, Germany.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 315–333, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany.

[BCS21] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. Sumcheck arguments and their applications. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 742–773, Virtual Event, August 16–20, 2021. Springer, Heidelberg, Germany.

[BCTV14a] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable zero knowledge via cycles of elliptic curves. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 276–294, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

[BCTV14b] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive zero knowledge for a von neumann architecture. In Kevin Fu and Jaeyeon Jung, editors, *USENIX Security 2014: 23rd USENIX Security Symposium*, pages 781–796, San Diego, CA, USA, August 20–22, 2014. USENIX Association.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany.

[BDFG21a] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo infinite: Proof-carrying data from additive polynomial commitments. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 649–680, Virtual Event, August 16–20, 2021. Springer, Heidelberg, Germany.

[BDFG21b] Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo infinite: Proof-carrying data from additive polynomial commitments. In *Annual International Cryptology Conference*, pages 649–680. Springer, 2021.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 10–24, Arlington, VA, USA, January 10–12, 2016. ACM-SIAM.

[BDN18] Dan Boneh, Manu Drijvers, and Gregory Neven. Compact multi-signatures for smaller blockchains. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 435–464, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Heidelberg, Germany.

[Ber14] Daniel J. Bernstein. A subfield-logarithm attack against ideal lattices: Computational algebraic number theory tackles lattice-based cryptography. The cr.yp.to blog, `https://blog.cr.yp.to/20140213-ideal.html`, 2014.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.

[BF11] Dan Boneh and David Mandell Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011: 14th International Conference on Theory and Practice of Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 1–16, Taormina, Italy, March 6–9, 2011. Springer, Heidelberg, Germany.

[BGH19] Sean Bowe, Jack Grigg, and Daira Hopwood. Halo: Recursive proof composition without a trusted setup. Cryptology ePrint Archive, Report 2019/1021, 2019. `https://eprint.iacr.org/2019/1021`.

[BGRR19] Aurélie Bauer, Henri Gilbert, Guénaël Renault, and Mélissa Rossi. Assessment of the key-reuse resilience of NewHope. In Mitsuru Matsui, editor, *Topics in Cryptology – CT-RSA 2019*, volume 11405 of *Lecture Notes in Computer Science*, pages 272–292, San Francisco, CA, USA, March 4–8, 2019. Springer, Heidelberg, Germany.

[BHH+19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 61–90, Nuremberg, Germany, December 1–5, 2019. Springer, Heidelberg, Germany.

[BISW17] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Lattice-based SNARGs and their application to more efficient obfuscation. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 247–277, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.

[BISW18] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Quasi-optimal SNARGs via linear multi-prover interactive proofs. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 222–255, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

[BK20] Dan Boneh and Sam Kim. One-time and interactive aggregate signatures from lattices. `https://crypto.stanford.edu/~skim13/agg_ots.pdf`, 2020.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 1–12, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Heidelberg, Germany.

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 441–469, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 1970.

[BLR+18] Shi Bai, Tancrède Lepoint, Adeline Roux-Langlois, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. *Journal of Cryptology*, 31(2):610–640, April 2018.

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 176–202, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[BMM+21] Benedikt Bünz, Mary Maller, Pratyush Mishra, Nirvan Tyagi, and Psi Vesely. Proofs for inner pairing products and applications. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part III*, volume 13092 of *Lecture Notes in Computer Science*, pages 65–97, Singapore, December 6–10, 2021. Springer, Heidelberg, Germany.

[BMRS20] Joseph Bonneau, Izaak Meckler, Vanishree Rao, and Evan Shapiro. Coda: Decentralized cryptocurrency at scale. *Cryptology ePrint Archive*, 2020.

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 273–289, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany.

[BR21] Katharina Boudgoust and Adeline Roux-Langlois. Compressed linear aggregate signatures based on module lattices. Cryptology ePrint Archive, Report 2021/263, 2021. `https://eprint.iacr.org/2021/263`.

[BS20] Nina Bindel and John M. Schanck. Decryption failure is more likely after success. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 206–225, Paris, France, April 15–17, 2020. Springer, Heidelberg, Germany.

[BSY18] Hanno Böck, Juraj Somorovsky, and Craig Young. Return of Bleichenbacher's oracle threat (ROBOT). In William Enck and Adrienne Porter Felt, editors, *USENIX Security 2018: 27th USENIX Security Symposium*, pages 817–849, Baltimore, MD, USA, August 15–17, 2018. USENIX Association.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 97–106, Palm Springs, CA, USA, October 22–25, 2011. IEEE Computer Society Press.

[BV17] Nir Bitansky and Vinod Vaikuntanathan. A note on perfect correctness by derandomization. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 592–606, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.

[CDH+20] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. NTRU. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/`

post-quantum-cryptography-standardization/round-3-submissions.

[CDPR16] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. Recovering short generators of principal ideals in cyclotomic rings. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 559–585, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[CDW17] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short stickelberger class relations and application to ideal-SVP. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 324–348, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.

[CF13] Dario Catalano and Dario Fiore. Vector commitments and their applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 55–72, Nara, Japan, February 26 – March 1, 2013. Springer, Heidelberg, Germany.

[CFG+20] Matteo Campanelli, Dario Fiore, Nicola Greco, Dimitris Kolonelos, and Luca Nizzardo. Incrementally aggregatable vector commitments and applications to verifiable decentralized storage. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 3–35, Daejeon, South Korea, December 7–11, 2020. Springer, Heidelberg, Germany.

[CG08] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 2008: 15th Conference on Computer and Communications Security*, pages 345–356, Alexandria, Virginia, USA, October 27–31, 2008. ACM Press.

[CGS14] Peter Campbell, Michael Groves, and Dan Shepherd. Soliloquy: A cautionary tale. Available at http://docbox.etsi.org/Workshop/2014/201410_CRYPTO/S07_Systems_and_Attacks/S07_Groves_Annex.pdf, 2014.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in*

Cryptology – EUROCRYPT 2010, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

[CLM23] Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. Lattice-based succinct arguments from vanishing polynomials - (extended abstract). 2023.

[CLMQ21] Yilei Chen, Alex Lombardi, Fermi Ma, and Willy Quach. Does fiat-shamir require a cryptographic hash function? In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part IV*, volume 12828 of *Lecture Notes in Computer Science*, pages 334–363, Virtual Event, August 16–20, 2021. Springer, Heidelberg, Germany.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 1–29, Nuremberg, Germany, December 1–5, 2019. Springer, Heidelberg, Germany.

[CMSZ22] A. Chiesa, F. Ma, N. Spooner, and M. Zhandry. Post-quantum succinct arguments: Breaking the quantum rewinding barrier. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 49–58, Los Alamitos, CA, USA, feb 2022. IEEE Computer Society.

[CPZ18] Alexander Chepurnoy, Charalampos Papamanthou, and Yupeng Zhang. Edrax: A cryptocurrency with stateless transaction validation. Cryptology ePrint Archive, Report 2018/968, 2018. `https://eprint.iacr.org/2018/968`.

[CRSS20] Valerio Cini, Sebastian Ramacher, Daniel Slamanig, and Christoph Striecks. CCA-secure (puncturable) KEMs from encryption with non-negligible decryption errors. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 159–190, Daejeon, South Korea, December 7–11, 2020. Springer, Heidelberg, Germany.

[DFGS15] Benjamin Dowling, Marc Fischlin, Felix Günther, and Douglas Stebila. A cryptographic analysis of the TLS 1.3 handshake protocol candidates. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *ACM CCS 2015: 22nd Conference on Computer and Communications Security*, pages 1197–1210, Denver, CO, USA, October 12–16, 2015. ACM Press.

[DGJ+21] David Derler, Kai Gellert, Tibor Jager, Daniel Slamanig, and Christoph Striecks. Bloom filter encryption and applications to efficient forward-secret 0-rtt key exchange. *J. Cryptol.*, 34(2):13, 2021.

[DGK20] Nir Drucker, Shay Gueron, and Dusan Kostic. On constant-time QC-MDPC decoders with negligible failure rate. In *CBCrypto*, volume 12087 of *Lecture Notes in Computer Science*, pages 50–79. Springer, 2020.

[DGKP21] Nir Drucker, Shay Gueron, Dusan Kostic, and Edoardo Persichetti. On the applicability of the fujisaki-okamoto transformation to the BIKE KEM. *Int. J. Comput. Math. Comput. Syst. Theory*, 6(4):364–374, 2021.

[DGNW20] Manu Drijvers, Sergey Gorbunov, Gregory Neven, and Hoeteck Wee. Pixel: Multi-signatures for consensus. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2093–2110. USENIX Association, August 2020.

[DHSS20] Yarkın Doröz, Jeffrey Hoffstein, Joseph H. Silverman, and Berk Sunar. MMSAT: A scheme for multimessage multiuser signature aggregation. Cryptology ePrint Archive, Report 2020/520, 2020. `https://eprint.iacr.org/2020/520`.

[DJSS18] David Derler, Tibor Jager, Daniel Slamanig, and Christoph Striecks. Bloom filter encryption and applications to efficient forward-secret 0-RTT key exchange. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 425–455, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 22–41, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg, Germany.

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 342–360, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

[DPW19] Léo Ducas, Maxime Plançon, and Benjamin Wesolowski. On the shortness of vectors to be found by the ideal-SVP quantum algorithm. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 322–351, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[DT18] Thomas Debris-Alazard and Jean-Pierre Tillich. Two attacks on rank metric code-based schemes: RankSign and an IBE scheme. In Thomas Peyrin and

Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 62–92, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Heidelberg, Germany.

[EEE20] Muhammed F Esgin, Oğuzhan Ersoy, and Zekeriya Erkin. Post-quantum adaptor signatures and payment channel networks. In *European Symposium on Research in Computer Security*, pages 378–397. Springer, 2020.

[ENS20] Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 259–288, Daejeon, South Korea, December 7–11, 2020. Springer, Heidelberg, Germany.

[FHS+17] Tomás Fabsic, Viliam Hromada, Paul Stankovski, Pavol Zajac, Qian Guo, and Thomas Johansson. A reaction attack on the QC-LDPC McEliece cryptosystem. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, pages 51–68, Utrecht, The Netherlands, June 26–28, 2017. Springer, Heidelberg, Germany.

[Fis05] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 152–168, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Heidelberg, Germany.

[Fis18] Ben Fisch. PoReps: Proofs of space on useful data. Cryptology ePrint Archive, Report 2018/678, 2018. `https://eprint.iacr.org/2018/678`.

[Fis19] Ben Fisch. Tight proofs of space and replication. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 324–348, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Heidelberg, Germany.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 105–111, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany.

[GGM14] Christina Garman, Matthew Green, and Ian Miers. Decentralized anonymous credentials. In *ISOC Network and Distributed System Security Symposium – NDSS 2014*, San Diego, CA, USA, February 23–26, 2014. The Internet Society.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 626–645, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

[GHJL17] Felix Günther, Britta Hale, Tibor Jager, and Sebastian Lauer. 0-RTT key exchange with full forward secrecy. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 519–548, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.

[GHPT17] Philippe Gaborit, Adrien Hauteville, Duong Hieu Phan, and Jean-Pierre Tillich. Identity-based encryption from codes with rank metric. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 194–224, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[GJN20] Qian Guo, Thomas Johansson, and Alexander Nilsson. A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 359–386, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

[GJS16] Qian Guo, Thomas Johansson, and Paul Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 789–815, Hanoi, Vietnam, December 4–8, 2016. Springer, Heidelberg, Germany.

[GJY19] Qian Guo, Thomas Johansson, and Jing Yang. A novel CCA attack using decryption errors against LAC. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 82–111, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.

[GKK+19] Lorenzo Grassi, Daniel Kales, Dmitry Khovratovich, Arnab Roy, Christian Rechberger, and Markus Schofnegger. Starkad and Poseidon: New hash functions for zero knowledge proof systems. Cryptology ePrint Archive, Report 2019/458, 2019. `https://eprint.iacr.org/2019/458`.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In Chris Umans, editor, *58th Annual Symposium on Foundations of Computer Science*, pages 612–621, Berkeley, CA, USA, October 15–17, 2017. IEEE Computer Society Press.

[GL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA, 1996.

[GLS+21] Alexander Golovnev, Jonathan Lee, Srinath Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and post-quantum SNARKs for R1CS. Cryptology ePrint Archive, Report 2021/1043, 2021. `https://eprint.iacr.org/2021/1043`.

[GM15] Matthew D. Green and Ian Miers. Forward secure asynchronous messaging from puncturable encryption. In *2015 IEEE Symposium on Security and Privacy*, pages 305–320, San Jose, CA, USA, May 17–21, 2015. IEEE Computer Society Press.

[GM17] Matthew Green and Ian Miers. Bolt: Anonymous payment channels for decentralized currencies. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 473–489, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press.

[GM18] Nicholas Genise and Daniele Micciancio. Faster Gaussian sampling for trapdoor lattices with arbitrary modulus. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 174–203, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

[GMNO18] Rosario Gennaro, Michele Minelli, Anca Nitulescu, and Michele Orrù. Lattice-based zk-SNARKs from square span programs. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 556–573, Toronto, ON, Canada, October 15–19, 2018. ACM Press.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[GR19] Alonso González and Carla Ràfols. Shorter pairing-based arguments under standard assumptions. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 728–757, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *28th Annual ACM Symposium on Theory of Computing*, pages 212–219, Philadephia, PA, USA, May 22–24, 1996. ACM Press.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 305–326, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[Gro21] Jonathan Gross. Practical snark based vdf, 2021. `https://zkproof.org/2021/11/24/practical-snark-based-vdf/`.

[GRWZ20] Sergey Gorbunov, Leonid Reyzin, Hoeteck Wee, and Zhenfei Zhang. Pointproofs: Aggregating proofs for multiple vector commitments. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020: 27th Conference on Computer and Communications Security*, pages 2007–2023, Virtual Event, USA, November 9–13, 2020. ACM Press.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[GVW15a] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B.

Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 503–523, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

[GVW15b] Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing*, pages 469–477, Portland, OR, USA, June 14–17, 2015. ACM Press.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 99–108, San Jose, CA, USA, June 6–8, 2011. ACM Press.

[GZB+19] Oscar Garcia-Morchon, Zhenfei Zhang, Sauvik Bhattacharya, Ronald Rietman, Ludo Tolhuizen, Jose-Luis Torre-Arce, Hayo Baan, Markku-Juhani O. Saarinen, Scott Fluhrer, Thijs Laarhoven, and Rachel Player. Round5. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions`.

[Ham19] Mike Hamburg. Three Bears. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions`.

[Har06] Ben Harris. RSA key exchange for the secure shell (SSH) transport layer protocol. *RFC*, 4432:1–8, 2006.

[HHK17a] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.

[HHK17b] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. Cryptology ePrint Archive, Report 2017/604, 2017. `https://eprint.iacr.org/2017/604`.

[HKSU20] Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas,

editors, *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 389–422, Edinburgh, UK, May 4–7, 2020. Springer, Heidelberg, Germany.

[HKW20] Susan Hohenberger, Venkata Koppula, and Brent Waters. Chosen ciphertext security from injective trapdoor functions. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 836–866, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

[HNP+03] Nick Howgrave-Graham, Phong Q. Nguyen, David Pointcheval, John Proos, Joseph H. Silverman, Ari Singer, and William Whyte. The impact of decryption failures on the security of NTRU encryption. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 226–246, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany.

[HPS96] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A new high speed public key cryptosystem, 1996. Draft Distributed at Crypto'96, available at `http://web.securityinnovation.com/hubfs/files/ntru-orig.pdf`.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288, 1998.

[HT98] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 408–423, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Heidelberg, Germany.

[HW15] Pavel Hubacek and Daniel Wichs. On the communication complexity of secure function evaluation with long output. In Tim Roughgarden, editor, *ITCS 2015: 6th Conference on Innovations in Theoretical Computer Science*, pages 163–172, Rehovot, Israel, January 11–13, 2015. Association for Computing Machinery.

[ISW21] Yuval Ishai, Hang Su, and David J. Wu. Shorter and faster post-quantum designated-verifier zkSNARKs from lattices. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021: 28th Conference on Computer and Communications Security*, pages 212–234, Virtual Event, Republic of Korea, November 15–19, 2021. ACM Press.

[JD11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum*

Cryptography - 4th International Workshop, PQCrypto 2011, pages 19–34, Tapei, Taiwan, November 29 – December 2 2011. Springer, Heidelberg, Germany.

[JKSS12] Tibor Jager, Florian Kohlar, Sven Schäge, and Jörg Schwenk. On the security of TLS-DHE in the standard model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 273–293, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.

[JSS12] Tibor Jager, Sebastian Schinzel, and Juraj Somorovsky. Bleichenbacher's attack strikes again: Breaking PKCS#1 v1.5 in XML encryption. In Sara Foresti, Moti Yung, and Fabio Martinelli, editors, *ESORICS 2012: 17th European Symposium on Research in Computer Security*, volume 7459 of *Lecture Notes in Computer Science*, pages 752–769, Pisa, Italy, September 10–12, 2012. Springer, Heidelberg, Germany.

[JZC+18] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 96–125, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

[JZM19] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 227–248, Chongqing, China, May 8–10, 2019. Springer, Heidelberg, Germany.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th Annual ACM Symposium on Theory of Computing*, pages 723–732, Victoria, BC, Canada, May 4–6, 1992. ACM Press.

[KM10] Neal Koblitz and Alfred Menezes. The brave new world of bodacious assumptions in cryptography. *Notices of the American Mathematical Society*, 57(3):357–365, 2010.

[KMS+16] Ahmed E. Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy*, pages 839–858, San Jose, CA, USA, May 22–26, 2016. IEEE Computer Society Press.

[KPW13] Hugo Krawczyk, Kenneth G. Paterson, and Hoeteck Wee. On the security of the TLS protocol: A systematic analysis. In Ran Canetti and Juan A.

Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 429–448, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

[KSS+20] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 703–728, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

[KW19] Venkata Koppula and Brent Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 671–700, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany.

[Laa15] Thijs Laarhoven. *Search problems in cryptography: From fingerprinting to lattice sieving*. PhD thesis, Eindhoven University of Technology, 2015.

[Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 169–189, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Heidelberg, Germany.

[Lis05] Moses Liskov. Updatable zero-knowledge databases. In Bimal K. Roy, editor, *Advances in Cryptology – ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 174–198, Chennai, India, December 4–8, 2005. Springer, Heidelberg, Germany.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact Knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006: 33rd International Colloquium on Automata, Languages and Programming, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155, Venice, Italy, July 10–14, 2006. Springer, Heidelberg, Germany.

[LM19] Russell W. F. Lai and Giulio Malavolta. Subvector commitments with application to succinct arguments. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 530–560, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[LM23] Russell W. F. Lai and Giulio Malavolta. Lattice-based timed-cryptography. In *CRYPTO 2023*, 2023.

[LMR19] Russell W. F. Lai, Giulio Malavolta, and Viktoria Ronge. Succinct arguments for bilinear group arithmetic: Practical structure-preserving cryptography. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 2057–2074, London, UK, November 11–15, 2019. ACM Press.

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 71–101, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Heidelberg, Germany.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 35–54, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

[LPSS14] San Ling, Duong Hieu Phan, Damien Stehlé, and Ron Steinfeld. Hardness of k-LWE and applications in traitor tracing. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 315–334, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

[LRY16] Benoît Libert, Somindu C. Ramanna, and Moti Yung. Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *ICALP 2016: 43rd International Colloquium on Automata, Languages and Programming*, volume 55 of *LIPIcs*, pages 30:1–30:14, Rome, Italy, July 11–15, 2016. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, June 2015.

[LY10] Benoît Libert and Moti Yung. Concise mercurial vector commitments and independent zero-knowledge sets with short proofs. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 499–517, Zurich, Switzerland, February 9–11, 2010. Springer, Heidelberg, Germany.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[Ma20] Fermi Ma. Quantum-secure commitments and collapsing hash functions. `https://www.cs.princeton.edu/~fermim/talks/collapse-binding.pdf`, April 2020.

[McE78] Robert J McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

[Mer79] Ralph Charles Merkle. *Secrecy, authentication, and public key systems.* Stanford university, 1979.

[Mic94] Silvio Micali. CS proofs (extended abstracts). In *35th Annual Symposium on Foundations of Computer Science*, pages 436–453, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press.

[Mic07] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Comput. Complex.*, 16(4):365–411, 2007.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press.

[MRK03] Silvio Micali, Michael O. Rabin, and Joe Kilian. Zero-knowledge sets. In *44th Annual Symposium on Foundations of Computer Science*, pages 80–91, Cambridge, MA, USA, October 11–14, 2003. IEEE Computer Society Press.

[MS04] Thom Mulders and Arne Storjohann. Certified dense linear system solving. *J. Symb. Comput.*, 37(4):485–510, 2004.

[NAB+19] Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions`.

[Nao90] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 128–136, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Heidelberg, Germany.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany.

[NDR+19] Hamid Nejatollahi, Nikil D. Dutt, Sandip Ray, Francesco Regazzoni, Indranil Banerjee, and Rosario Cammarota. Post-quantum lattice-based cryptography implementations: A survey. *ACM Comput. Surv.*, 51(6):129:1–129:41, 2019.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*, pages 427–437, Baltimore, MD, USA, May 14–16, 1990. ACM Press.

[Pat96] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48, Saragossa, Spain, May 12–16, 1996. Springer, Heidelberg, Germany.

[PFH+20] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252, Berkeley, CA, USA, May 19–22, 2013. IEEE Computer Society Press.

[PHS19] Alice Pellet-Mary, Guillaume Hanrot, and Damien Stehlé. Approx-SVP in ideal lattices with pre-processing. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 685–716, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[Pie19] Krzysztof Pietrzak. Simple verifiable delay functions. In Avrim Blum, editor, *ITCS 2019: 10th Innovations in Theoretical Computer Science Conference*, volume 124, pages 60:1–60:15, San Diego, CA, USA, January 10–12, 2019. LIPIcs.

[PPS21] Chris Peikert, Zachary Pepin, and Chad Sharp. Vector and functional commitments from lattices. In Kobbi Nissim and Brent Waters, editors, *TCC 2021: 19th Theory of Cryptography Conference, Part III*, volume 13044 of *Lecture Notes in Computer Science*, pages 480–511, Raleigh, NC, USA, November 8–11, 2021. Springer, Heidelberg, Germany.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 145–166, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany.

[PS21] Alice Pellet-Mary and Damien Stehlé. On the hardness of the NTRU problem. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 3–35, Singapore, December 6–10, 2021. Springer, Heidelberg, Germany.

[PST20] Christian Paquin, Douglas Stebila, and Goutam Tamvada. Benchmarking post-quantum cryptography in TLS. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 72–91, Paris, France, April 15–17, 2020. Springer, Heidelberg, Germany.

[PXWC21] Yanbin Pan, Jun Xu, Nick Wadleigh, and Qi Cheng. On the ideal shortest vector problem over random rational primes. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 559–583, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany.

[RAD+78] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.

[RGG+19] Eyal Ronen, Robert Gillham, Daniel Genkin, Adi Shamir, David Wong, and Yuval Yarom. The 9 lives of Bleichenbacher's CAT: New cache ATtacks on TLS implementations. In *2019 IEEE Symposium on Security and Privacy*, pages 435–452, San Francisco, CA, USA, May 19–23, 2019. IEEE Computer Society Press.

[SAB+19] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions`.

[SE94] Claus-Peter Schnorr and Michael Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66:181–199, 1994.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pages 124–134, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press.

[SS11] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 27–47, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany.

[SSPB19] Simona Samardjiska, Paolo Santini, Edoardo Persichetti, and Gustavo Banegas. A reaction attack against cryptosystems based on LRPC codes. In Peter Schwabe and Nicolas Thériault, editors, *Progress in Cryptology - LATINCRYPT 2019: 6th International Conference on Cryptology and Information Security in Latin America*, volume 11774 of *Lecture Notes in Computer Science*, pages 197–216, Santiago, Chile, October 2–4, 2019. Springer, Heidelberg, Germany.

[SSS+20] Shifeng Sun, Amin Sakzad, Ron Steinfeld, Joseph K. Liu, and Dawu Gu. Public-key puncturable encryption: Modular and compact constructions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 12110 of *Lecture Notes in Computer Science*, pages 309–338, Edinburgh, UK, May 4–7, 2020. Springer, Heidelberg, Germany.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635, Tokyo, Japan, December 6–10, 2009. Springer, Heidelberg, Germany.

[SSW20a] Peter Schwabe, Douglas Stebila, and Thom Wiggers. Post-quantum TLS without handshake signatures. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020: 27th Conference on Computer and Communications Security*, pages 1461–1480, Virtual Event, USA, November 9–13, 2020. ACM Press.

[SSW20b] Peter Schwabe, Douglas Stebila, and Thom Wiggers. Post-quantum TLS without handshake signatures. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020: 27th Conference on Computer and Communications Security*, pages 1461–1480, Virtual Event, USA, November 9–13, 2020. ACM Press.

[SV19] Nicolas Sendrier and Valentin Vasseur. On the decoding failure rate of QC-MDPC bit-flipping decoders. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 404–416, Chongqing, China, May 8–10, 2019. Springer, Heidelberg, Germany.

[SV20] Nicolas Sendrier and Valentin Vasseur. About low DFR for QC-MDPC decoding. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 20–34, Paris, France, April 15–17, 2020. Springer, Heidelberg, Germany.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

[Unr15] Dominique Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6):49:1–49:76, 2015.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

[Val08] Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 1–18, San Francisco, CA, USA, March 19–21, 2008. Springer, Heidelberg, Germany.

[vGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany.

[WW23a] Hoeteck Wee and David J. Wu. Lattice-based functional commitments: Fast verification and cryptanalysis. private communication, May 2023.

[WW23b] Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. In *Advances in Cryptology – EUROCRYPT 2023, Part III*, Lecture Notes in Computer Science, pages 385–416. Springer, Heidelberg, Germany, June 2023.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In Chris Umans, editor, *58th Annual Symposium on Foundations of Computer Science*, pages 600–611, Berkeley, CA, USA, October 15–17, 2017. IEEE Computer Society Press.

[YAZ+19] Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 147–175, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[ZCZ16] Jiang Zhang, Yu Chen, and Zhenfeng Zhang. Programmable hash functions from lattices: Short signatures and IBEs with small key sizes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 303–332, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.

APPENDIX A

Appendix to Chapter 2

A.1 Omitted Definitions

A.1.1 QROM Definitions and Lemmas

We recall various lemmas that we require for the QROM proofs. While in the ROM, the simulator always learns x and $H(x)$ if the adversary tries to learn any information on $H(x)$, the situation in the QROM is not as simple. Measuring or recording queries might collapse the adversary's quantum state and change its behavior. The simulator can however learn queries under certain conditions using the "one-way to hiding" (O2H) technique [Unr15].

In the following we consider two quantum-accessible oracles $G, H\colon X \to Y$, but they do not have to be random oracles. Let's assume that G and H only differ in some small set $S \subset X$, i.e. $G_{|X\setminus S} = H_{|X\setminus S}$. Now consider an algorithm $\mathcal{A}$ that makes at most q queries to G or H. Since queries can be made in parallel, suppose that the maximum number of sequential invocations of the oracles, the depth, is at most $d \leq q$. Now, for some input z, the O2H technique gives a way for the simulator to find some $x \in S$ if $\mathcal{A}^G(z)$ behaves differently from $\mathcal{A}^H(z)$.

The first lemma is the original one-way to hiding lemma which first appeared in [Unr15]. We use the formulation from [BHH+19], i.e. by conditioning the probabilities on a classical event Ev.[1]

Lemma A.1.1 (Thm. 3 [AHU19])**.** *Let $G, H\colon X \to Y$ be random functions, let z be a random value, and let $S \subset X$ be a random set such that $G_{|X\setminus S} = H_{|X\setminus S}$. Furthermore, let $\mathcal{A}^H$ be a quantum oracle algorithm which queries H with depth at most d. Let Ev be an arbitrary classical event. Define an oracle algorithm $B^H(z)$ as follows: pick $i \leftarrow [d]$*

[1] Throughout this section (G, H, S, z) may have an arbitrary joint distribution.

and run $\mathcal{A}^H(z)$ until just before its i-th round of queries of H. Measure all query input registers in the computational basis, and output the set T of measurement outcomes. Define

$$\begin{aligned} P_{left} &= \Pr[\mathcal{A}^H(z) : \mathsf{Ev}], \\ P_{right} &= \Pr[\mathcal{A}^G(z) : \mathsf{Ev}], \\ P_{guess} &= \Pr[T \leftarrow B^H(z) : S \cap T \neq \emptyset]. \end{aligned}$$

Then

$$|P_{left} - P_{right}| \leq 2d\sqrt{P_{guess}}, \quad \text{and} \quad \left|\sqrt{P_{left}} - \sqrt{P_{right}}\right| \leq 2d\sqrt{P_{guess}}.$$

The same results holds with $B^G(z)$ in the definition of P_{guess}.

Next up, we move on to the semi-classical O2H. For that we need punctured oracles [AHU19] which measure whether the input is in a set S.

Definition A.1.1. *Let $H\colon X \to Y$ be any function, and let $S \subset X$ be a set. The oracle $H \setminus S$ takes as input a value x. It first computes whether $x \in S$ into an auxiliary qubit p and measures p. Then it runs $H(x)$ and returns the result. Let Find be the event that any of the measurements of p returns 1.*

We recall the "puncturing is effective" lemma, the "semi-classical one-way to hiding" lemma, as well as the "search in the semi-classical oracle" lemma.

Lemma A.1.2 (Lemma 1 [AHU19])**.** *Let $G, H\colon X \to Y$ be random functions, let z be a random value, and let $S \subset X$ be a random set such that $G_{|X\setminus S} = H_{|X\setminus S}$. Furthermore, let $\mathcal{A}^H$ be a quantum oracle algorithm. Let Ev be an arbitrary classical event. Then*

$$\Pr[\mathcal{A}^{H\setminus S}(z) : \mathsf{Ev} \wedge \neg\mathsf{Find}] = \Pr[\mathcal{A}^{G\setminus S}(z) : \mathsf{Ev} \wedge \neg\mathsf{Find}].$$

Lemma A.1.3 (Thm. 1 [AHU19])**.** *Let $G, H\colon X \to Y$ be random functions, let z be a random value, and let $S \subset X$ be a random set such that $G_{|X\setminus S} = H_{|X\setminus S}$. Furthermore, let $\mathcal{A}^H$ be a quantum oracle algorithm which queries H with depth at most d. Let Ev be an arbitrary classical event. Define*

$$\begin{aligned} P_{left} &= \Pr[\mathcal{A}^H(z) : \mathsf{Ev}], \\ P_{right} &= \Pr[\mathcal{A}^G(z) : \mathsf{Ev}], \\ P_{find} &= \Pr[\mathcal{A}^{H\setminus S}(z) : \mathsf{Find}] = \Pr[\mathcal{A}^{G\setminus S}(z) : \mathsf{Find}]. \end{aligned}$$

Then

$$|P_{left} - P_{right}| \leq 2\sqrt{dP_{find}} \text{ and} \quad \left|\sqrt{P_{left}} - \sqrt{P_{right}}\right| \leq 2\sqrt{dP_{find}}.$$

The theorem also holds with bound $\sqrt{(d+1)P_{find}}$ *for the following alternative definitions of* P_{right}*:*

$$P_{right} = \Pr[\mathcal{A}^{H\setminus S}(z) : \mathsf{Ev}],$$
$$P_{right} = \Pr[\mathcal{A}^{H\setminus S}(z) : \mathsf{Ev} \wedge \neg\mathsf{Find}] = \Pr[\mathcal{A}^{G\setminus S}(z) : \mathsf{Ev} \wedge \neg\mathsf{Find}],$$
$$P_{right} = \Pr[adv A^{H\setminus S}(z) : \mathsf{Ev} \vee \mathsf{Find}] = \Pr[adv A^{G\setminus S}(z) : \mathsf{Ev} \vee \mathsf{Find}].$$

Lemma A.1.4 (Thm. 2, Cor. 1 [AHU19])**.** *Let* $H\colon X \to Y$ *be a random function, let* z *be a random value, and let* $S \subset X$ *be a random set. Let* $\mathcal{A}^H$ *be a quantum oracle algorithm which queries* H *at most* q *times with depth at most* d*. Let* $B^H(z)$ *and* P_{guess} *be defined as in Lemma A.1.1. Then*

$$\Pr[\mathcal{A}^{H\setminus S}(z) : \mathsf{Find}] \leq 4dP_{guess}.$$

In particular, if for each $x \in X$*,* $\Pr[x \in S] \leq \varepsilon$ *(conditioned on* z*, on other oracles* $\mathcal{A}$ *has access to, and on other outputs of* H*), then*

$$\Pr[\mathcal{A}^{H\setminus S}(z) : \mathsf{Find}] \leq 4q\varepsilon.$$

A.2 Omitted Proofs

A.2.1 Proof of Theorem 2.3.9

Proof. The correctness claim follows directly from the analysis in Section 2.3.4: consider an adversary A in the random oracle model. We can assume that it makes at most q_{G} (distinct) queries and that q_{G} is a multiple of the number ℓ of Π ciphertexts that form a Π' one. Let $q'_{\mathsf{G}} = q_{\mathsf{G}}/\ell$, $h \in [q'_{\mathsf{G}}]$, and let

$$\mathsf{G}(M_1,1),\ldots,\mathsf{G}(M_1,\ell),\ldots,\mathsf{G}(M_{q'_{\mathsf{G}}},1),\ldots,\mathsf{G}(M_{q'_{\mathsf{G}}},\ell),$$

be the queries to G. We call a tuple $\mathsf{G}(M_h,1),\ldots,\mathsf{G}(M_h,\ell)$ *problematic* iff it exhibits a correctness error in Π' (in the sense that $\Pi'.\mathsf{Dec}(\mathsf{sk},\Pi'.\mathsf{Enc}(\mathsf{pk},M_h)) \neq M_h$). By Section 2.3.4, we know that, for any key pair $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$ satisfying the δ-correctness bound, each tuple $\mathsf{G}(M_h,1),\ldots,\mathsf{G}(M_h,\ell)$ is *problematic* with probability at most $\ell\delta^\ell$. Hence, a union bound shows that at least a tuple $\mathsf{G}(M_h,1),\ldots,\mathsf{G}(M_h,\ell)$ is *problematic* is at most $q'_{\mathsf{G}} \cdot \ell\delta^\ell = \frac{q_{\mathsf{G}}}{\ell} \cdot \ell\delta^\ell = q_{\mathsf{G}} \cdot \delta^\ell$.

Now, we argue the security and therefore let B be an adversary against the OW-PCA security of Π' issuing at most q_{G} queries to G and at most q_P queries to Pco. We proceed with a sequence of games. Let $\mathsf{Adv}_{B,j}$ be the advantage of B in Game j.

Game G_0: This is the original OW-PCA game, where we simulate the random oracle queries $\mathsf{G}(M,i)$ as follows: if there exits r s.t. $(M,i,r) \in \mathcal{Q}_{\mathsf{G}}$, then return $\mathsf{G}(M,i) := r$. Otherwise choose $r \leftarrow \Pi.\mathcal{R}$, set $\mathcal{Q}_{\mathsf{G}} := \mathcal{Q}_{\mathsf{G}} \cup \{(M,i,r)\}$ and return $\mathsf{G}(M,i) := r$. Consequently, we have

$$\mathsf{Adv}_{B,0} = \mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}pca}}_{\Pi',B}(\lambda).$$

Game G_1: In game G_1 we replace the plaintext checking oracle $\textsc{Pco}(M, \mathsf{ctxt})$ by a simulation that does not check whether $M = M'$, where $M' = \mathsf{Dec}(\mathsf{sk}, C)$, anymore but simply computes $\mathsf{ctxt}_i := \Pi.\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M, i))$ for all $i \in [\ell]$ (i.e., re-encrypts M) and checks if $(C_1, \ldots, C_\ell) = \mathsf{ctxt}$. We claim

$$|\mathsf{Adv}_{B,1} - \mathsf{Adv}_{B,0}| \leq \epsilon + \left(\frac{q_\mathsf{G}}{\ell} + q_P\right) \cdot \ell\delta^\ell. \tag{A.1}$$

To show eq. (A.1), observe that at most q_G (distinct) queries to G are made in both G_0 and G_1. We can assume that q_G is a multiple of the number ℓ of Π ciphertexts that form a Π' one. Let $q'_\mathsf{G} = q_\mathsf{G}/\ell$, $h \in [q'_\mathsf{G}]$, and let

$$\mathsf{G}(M_1, 1), \ldots, \mathsf{G}(M_1, \ell), \ldots, \mathsf{G}(M_{q'_\mathsf{G}}, 1), \ldots, \mathsf{G}(M_{q'_\mathsf{G}}, \ell),$$

be the queries to G.

Let Bad be the event where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$ is one of those key pairs not satisfying the δ-correctness bound, and call a tuple $\mathsf{G}(M_h, 1), \ldots, \mathsf{G}(M_h, \ell)$ *problematic* iff it exhibits a correctness error in Π' (in the sense that $\Pi'.\mathsf{Dec}(\mathsf{sk}, \Pi'.\mathsf{Enc}(\mathsf{pk}, M_h)) \neq M_h$). Clearly, if B makes a *problematic* tuple query, then there exists an adversary D that triggers a correctness error for Π'. From what was discussed in the analysis of the correctness error of the compiler T^*, we know that conditioned on $\neg\mathsf{Bad}$, each tuple $\mathsf{G}(M_h, 1), \ldots, \mathsf{G}(M_h, \ell)$ is *problematic* with probability at most $\ell\delta^\ell$. Hence, a union bound shows that the probability that at least a tuple $\mathsf{G}(M_h, 1), \ldots, \mathsf{G}(M_h, \ell)$, $h \in [q'_\mathsf{G}]$, is *problematic* is at most $q'_\mathsf{G} \cdot \ell\delta^\ell = \frac{q_\mathsf{G}}{\ell} \cdot \ell\delta^\ell = q_\mathsf{G} \cdot \delta^\ell$. However, conditioned on $\neg\mathsf{Bad}$ and on the event that no tuple query is problematic, games G_0 and G_1 proceed identically. Indeed, the two games only differ if B submits a $\textsc{Pco}$ query (M, ctxt) together with G queries $(M, 1), \ldots, (M, \ell)$ such that $\mathsf{G}(M, 1), \ldots, \mathsf{G}(M, \ell)$ is *problematic* and $C = \Pi'.\mathsf{Enc}(\mathsf{pk}, M)$. Consequently, we have

$$|\mathsf{Adv}_{B,1} - \mathsf{Adv}_{B,0}| \leq \epsilon + \left(\frac{q_\mathsf{G}}{\ell} + q_P\right) \cdot \ell\delta^\ell,$$

as claimed.

Game G_2: In Game G_2, we consider event $\mathcal{E}$, which we define to be a query (M, i) to G for challenge message M and $i \in [\ell]$, or equivalently $(M, \cdot, \cdot) \in \mathcal{Q}_\mathsf{G}$. We abort if event $\mathcal{E}$ happens and due to the difference lemma we have

$$|\mathsf{Adv}_{B,2} - \mathsf{Adv}_{B,1}| \leq \Pr[\mathcal{E}].$$

Now, we can construct an adversary against the OW-CPA security of Π in that by obtaining a challenge ciphertext C for a unknown random M we provide $(\mathsf{pk}, \mathsf{ctxt})$ to the adversary B and we forward the output M' of B to the OW-CPA challenger. Using Lemma 2.2.1, relating OW-CPA and IND-CPA, we thus obtain:

$$\mathsf{Adv}_{B,2} = \frac{1}{|\mathcal{M}|} + \mathsf{Adv}^{\text{n-pke-ind-cpa}}_{\Pi,B}(\lambda).$$

Finally, we bound $\Pr[\mathcal{E}]$ and construct an ℓ-IND-CPA adversary against PKE Π that wins if event $\mathcal{E}$ happens in Game G_2. Therefore, we choose $(M_0, M_1) \leftarrow \mathcal{M}^2$ and send it ℓ times to the ℓ-IND-CPA challenger obtaining $(\mathsf{ctxt}_{b,1}, \ldots, \mathsf{ctxt}_{b,\ell})$ for M_b with unknown bit b and forward $(\mathsf{pk}, (\mathsf{ctxt}_{b,1}, \ldots, \mathsf{ctxt}_{b,\ell}))$ to B simulating its view in Game G_2. Now we consider event $\mathcal{B}$ being the event that B does query (M_{b-1}, j) for some arbitrary $j \in [\ell]$ to G. Since M_{b-1} is chosen uniformly random from $\mathcal{M}$ and independent of B's view, we have $\Pr[\mathcal{B}] \leq \frac{q_\mathsf{G}}{|\mathcal{M}|}$. For the remainder let us assume that event $\mathcal{B}$ did not happen. Note that if $\mathcal{E}$ happens, then B queried the random oracle G on M_b for some $i \in [\ell]$ and thus $b = b'$. If $\mathcal{E}$ does not happen, then B did neither query M_b on G nor M_{b-1}, we choose a random bit b' and thus $\Pr[b = b'] = \frac{1}{2}$. Overall, we then have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{n\text{-}pke\text{-}ind\text{-}cpa}}_{\Pi,B}(\lambda) + \frac{q_\mathsf{G}}{|\mathcal{M}|} &\geq \left|\Pr[b = b'] - \frac{1}{2}\right| \\ &= \left|\Pr[\mathcal{E}] + \frac{1}{2} \cdot \Pr[\neg\mathcal{E}] - \frac{1}{2}\right| \\ &= \frac{1}{2} \cdot \Pr[\mathcal{E}].\end{aligned}$$

Taking all together and using Theorem 2.2.3 yields the desired bound. □

A.2.2 Proof of Theorem 2.3.10

Proof. Let $\mathcal{A}_1$ be the same as $\mathcal{A}$ but after choosing an output M, compute and discard $\mathsf{G}(M, i)$ for all $i \in [\ell]$. Hence, it makes at most $q_\mathsf{G} + \ell$ queries at depth at most $d + \ell$. Thus, returning the correct M will always count as a Find later in the proof (c.f. Definition A.1.1). The two algorithms have the same OW-CPA-advantage of $\Pi' = \mathsf{T}^\star[\Pi, \mathsf{G}, \ell]$.

As Bindel et al. we show a slightly stronger result by constructing an IND-KPA adversary B with ℓ challenge ciphertexts, i.e. the adversary is given a tuple $(\mathsf{pk}, M_0, M_1, \mathsf{ctxt}_1, \ldots, \mathsf{ctxt}_\ell)$ with $\mathsf{ctxt}_i = \mathsf{Enc}(\mathsf{pk}, M_b; r_i)$ and needs to determine b. The algorithm B creates a fresh random oracle G and runs $\mathcal{A}_1^{\mathsf{G}\setminus F}$ with $F = \{(M_b, i)_{b\in\{0,1\}, i\in[\ell]}\}$. Now assume that Find occurs, B measures whether the query was (M_0, i) or (M_1, i) for some i and returns the corresponding b. If the oracle is queried on both (M_0, i) or (M_1, i') or Find does not occur, B guesses b at random.

Let G' be the oracle such that $\mathsf{G}'(M_b, i) = r_i$ and $\mathsf{G}'(M, i) = \mathsf{G}(M, i)$ for all other messages. G' is unknown to B, but we can still analyze $\mathcal{A}$'s behavior when run with G' instead of G. By construction, $\mathcal{A}_1^{\mathsf{G}'\setminus F}$ cannot return m_b without causing Find. Hence, by Lemma A.1.3,

$$\begin{aligned}\sqrt{\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi',\mathcal{A}}(\lambda)} &= \sqrt{\Pr[m_b \leftarrow \mathcal{A}^{\mathsf{G}'}]} \\ &= \left|\sqrt{\Pr[m_b \leftarrow \mathcal{A}^{\mathsf{G}'}]}\right| - \underbrace{\sqrt{\Pr[m_b \leftarrow \mathcal{A}_1^{\mathsf{G}'\setminus F} \wedge \neg\mathsf{Find}]}}_{=0} \\ &\leq \sqrt{(d + \ell + 1)\Pr[\mathcal{A}_1^{\mathsf{G}'\setminus F} : \mathsf{Find}]}.\end{aligned}$$

By Lemma A.1.2, we obtain

$$\begin{aligned}\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\Pi',\mathcal{A}}(\lambda) &\leq (d+\ell+1)\cdot \Pr[\mathcal{A}_1^{\mathsf{G}'\setminus F} : \mathsf{Find}] \\ &= (d+\ell+1)\cdot \Pr[\mathcal{A}_1^{\mathsf{G}\setminus F} : \mathsf{Find}] = (d+\ell+1)\cdot \Pr[B : \mathsf{Find}].\end{aligned}$$

We now split the event Find as $\mathsf{Find}_b \vee \mathsf{Find}_{\neg b}$. In both cases Find occurs and in the first case M_b is measured whereas in the second $M_{\neg b}$ is measured. Then $\mathsf{Adv}^{\mathsf{n\text{-}pke\text{-}ind\text{-}kpa}}_{\Pi,B}(\lambda) = |\Pr[\mathsf{Find}_b] - \Pr[\mathsf{Find}_{\neg b}]|$.

Since B measures M whenever Find occurs, we can view $\mathsf{G}\setminus F$ as $\mathsf{G}''\setminus\{(M_{\neg b},i)_{i\in[\ell]}\} = (\mathsf{G}\setminus\{(M_b,i)_{i\in[\ell]}\})\setminus\{(M_{\neg b},i)_{i\in[\ell]}\}$. Since $\mathcal{A}$ has no information about $M_{\neg b}$ except from puncturing, it holds that for any M that $\Pr[\mathcal{A}^{\mathsf{G}''} : M\in\{M_{\neg b}\}] = \frac{1}{|\mathcal{M}|}$. Thus, by Lemma A.1.4, we have

$$\Pr[B : \mathsf{Find}_{\neg b}] \leq \frac{4\cdot(q_{\mathsf{G}}+1)}{|\mathcal{M}|}.$$

Consequently,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{n\text{-}pke\text{-}ind\text{-}kpa}}_{\Pi,B}(\lambda) &= |\Pr[B : \mathsf{Find}_b] - \Pr[B : \mathsf{Find}_{\neg b}]| \\ &\geq \Pr[B : \mathsf{Find}] - 2\Pr[B : \mathsf{Find}_{\neg b}] \geq \Pr[B : \mathsf{Find}] - \frac{8\cdot(q_{\mathsf{G}}+1)}{|\mathcal{M}|}.\end{aligned}$$

Since $\mathsf{Adv}^{\mathsf{n\text{-}pke\text{-}ind\text{-}kpa}}_{\Pi,B}(\lambda) \leq \mathsf{Adv}^{\mathsf{n\text{-}pke\text{-}ind\text{-}cpa}}_{\Pi,B}(\lambda) \leq \ell\cdot\mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,B}(\lambda)$ (by Theorem 2.2.3), we conclude with

$$\mathsf{Adv}^{\mathsf{pke\text{-}ow\text{-}cpa}}_{\mathsf{T}^\star[\Pi,\mathsf{G},\ell],\mathcal{A}}(\lambda) \leq (d+\ell+1)\cdot\left(\ell\cdot\mathsf{Adv}^{\mathsf{pke\text{-}ind\text{-}cpa}}_{\Pi,B}(\lambda) + \frac{8\cdot(q_{\mathsf{G}}+1)}{|\mathcal{M}|}\right).$$

□

A.2.3 Proof of Theorem 2.3.1

Proof. Let $(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{KGen}(\lambda)$. For $M\in\mathcal{M}$, define the set of coins such that decryption of M will succeed as

$$Y_M = \{r\in\mathcal{R} \mid \mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},M;r)) = M\}.$$

Define a new random oracle G' as $\mathsf{G}'(M,i) = \mathsf{G}(M,i)$ if $\mathsf{G}(M,i)\in Y_M$, $\mathsf{G}'(M,i)\leftarrow\mathcal{R}$ if $Y_M = \emptyset$, and $\mathsf{G}'(M,i)\leftarrow Y_M$ otherwise. Thus G' is uniformly random in the space of oracles where decryption succeeds if possible and G' is independent of the behavior of messages and ciphertexts for $\mathsf{T}^\star[\Pi,\mathsf{G},\ell]$ which do not decrypt correctly. Define the failure probability for a fixed key pair and G' as

$$\delta' = \max_{M\in\mathcal{M}} \Pr[\mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},M)) \neq M].$$

Additionally, define the event $\mathsf{DblFail}$ as the case that ctxt is the encryption of two messages M_1 and M_2 such that decryption fails, i.e. $\mathsf{Dec}(\mathsf{sk},\mathsf{ctxt})\notin\{M_1,M_2\}$. Define

$\varepsilon' = \Pr[\mathsf{DblFail}]$. Both δ' and ε' are independent of G'. We denote the event that $\mathcal{A}$ wins the FFC game as Fail and define $\mathsf{Ev} = \mathsf{Fail} \wedge \neg\mathsf{DblFail}$. By Lemma A.1.1:

$$\left|\sqrt{Pr[\mathcal{A}^{\mathsf{G}}(\mathsf{pk}) : \mathsf{Ev}]} - \sqrt{\Pr[\mathcal{A}^{\mathsf{G}'}(\mathsf{pk}) : \mathsf{Ev}]}\right| \leq 2 \cdot d \cdot \sqrt{P_{\text{guess}}}.$$

Conditioned on G', for each m we have that $\mathsf{G}(m,i) \neq \mathsf{G}'(m,i)$ for all $i \in [\ell]$ with probability at most $(\delta')^\ell$. Hence, with $\frac{q_{\mathsf{G}}}{d}$ guesses (in expectation), we have that

$$2 \cdot d \cdot \sqrt{P_{\text{guess}}} \leq \sqrt{4 \cdot d^2 \cdot P_{\text{guess}}} \leq \sqrt{4 \cdot d \cdot q_{\mathsf{G}} \cdot (\delta')^\ell}.$$

For a ciphertext ctxt define

$$p_1(c) = \Pr[\exists! M \in \mathcal{M}, \forall i \in [\ell] : \mathsf{ctxt}_i = \mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(m,i)) \wedge \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt}_i) \neq M].$$

Note that if M exists but is not unique, then $\mathsf{DblFail}$ occurs. Let $p_1 = \max_c p_1(c)$. Since p_1 is independent of G', we have

$$\Pr[\mathcal{A}^{\mathsf{G}'}(\mathsf{pk}) : \mathsf{Ev}] \leq q_L \cdot p_1.$$

From Lemma A.1.1, we obtain $p_1 \leq (\delta')^\ell + \sqrt{3(\varepsilon')^\ell}$. From the Cauchy-Schwarz corollary we obtain:

$$\begin{aligned}\Pr[\mathcal{A}^{\mathsf{G}}(\mathsf{pk}) : \mathsf{Ev}] &\leq \sqrt{4 \cdot d \cdot q_{\mathsf{G}} \cdot (\delta')^\ell} + \sqrt{q_L \cdot \left((\delta')^\ell + \sqrt{3 \cdot (\varepsilon')^\ell}\right)} \\ &\leq \sqrt{\left((4 \cdot d + 1) \cdot (\delta')^\ell + \sqrt{3 \cdot (\varepsilon')^\ell}\right) \cdot (q_{\mathsf{G}} + q_L)}.\end{aligned}$$

Finally, note that $\delta = E[\delta' : \mathsf{pk}, \mathsf{G}]$ and $\varepsilon \leq E[(\varepsilon')^\ell : \mathsf{pk}, \mathsf{G}]$. By Jensen's inequality it holds that $\sqrt{\varepsilon} \leq E[\sqrt{(\varepsilon')^\ell} : \mathsf{pk}, \mathsf{G}]$, and thus

$$\mathsf{Adv}^{\mathsf{pke\text{-}ffc}}_{\mathsf{T}^\star[\Pi,\mathsf{G},\ell],\mathcal{A}}(\lambda) \leq \left((4 \cdot d + 1) \cdot \delta^\ell + \sqrt{3 \cdot \varepsilon}\right) \cdot (q_{\mathsf{G}} + q_L) + \varepsilon.$$

□

A.2.4 Proof of Theorem 2.5.10

Theorem A.2.1. *If a BFKEM* BFKEM *is* BFKEM-IND-CPA*-secure with the (extended) correctness, separable randomness, publicly-checkable puncturing, and γ-spreadness properties, then* BFKEM′ *is* BFKEM-IND-CCA*-secure. Concretely, for any PPT adversary $\mathcal{A}$ making at most $q_{\mathsf{G}} = q_{\mathsf{G}}(\lambda)$ queries to the random oracle* G *and negligible $\delta = \delta(\lambda)$, there is a distinguisher $\mathcal{D}$ in the* BFKEM-IND-CPA*-security experiment such that*

$$\mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cca}}_{\mathsf{BFKEM}',\mathcal{A}}(\lambda, m, k) \leq \mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cpa}}_{\mathsf{BFKEM},\mathcal{D}}(\lambda, m, k) + 3 \cdot \delta + \frac{q_{\mathsf{G}}}{2^\gamma}. \tag{A.2}$$

Decapsulation-Oracle: $\mathsf{Decaps}'(\mathsf{sk}_i, \mathsf{ctxt})$

0 :	**if** $\mathsf{CheckPunct}(\mathsf{pk}, \mathcal{L}, \mathsf{ctxt}) = \bot$ **return** $\bot$	// G_3 - G_7
1 :	$\mathsf{k} \leftarrow \mathsf{Decaps}(\mathsf{sk}_i, \mathsf{ctxt})$	// G_0 - G_3
2 :	$\mathsf{k} \leftarrow \mathsf{Decaps}(\mathsf{sk}_0, \mathsf{ctxt})$	// G_4 - G_6
3 :	**if** $\mathsf{k} = \bot$ **return** $\bot$	// G_0
4 :	**if** $(\mathsf{k}, \mathsf{G}(\mathsf{k})) \notin \mathcal{L}$ **return** $\bot$	// G_1 - G_4
5 :	$(r, \mathsf{k}') := \mathsf{G}(\mathsf{k})$	// G_0 - G_1
6 :	read (unique) $(\mathsf{k}, (r, \mathsf{k}'))$ from $\mathcal{L}$	// G_2 - G_4
7 :	**if** $(\mathsf{ctxt}, \mathsf{k}) \neq \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k}))$ **return** $\bot$	// G_0 - G_4
8 :	**return** k	// G_0 - G_4
9 :	**if** $(\mathsf{k}, (r, \mathsf{k}')) \notin \mathcal{L}$ and $(\mathsf{ctxt}, \mathsf{k}) = \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k}))$ **return** $\bot$	// G_5
10 :	**return** k' such that $(\mathsf{k}, (r, \mathsf{k}')) \in \mathcal{L}$ and $(\mathsf{ctxt}, \mathsf{k}) = \mathsf{Encaps}(\mathsf{pk}, (r, \mathsf{k}))$	// G_5
11 :	**if** $(\hat{\mathsf{k}}, (\hat{r}, \hat{\mathsf{k}}')) \notin \mathcal{L}$ and $(\mathsf{ctxt}, \hat{\mathsf{k}}) = \mathsf{Encaps}(\mathsf{pk}; (\hat{r}, \hat{\mathsf{k}}))$ **return** $\bot$	// G_6 - G_7
12 :	**return** $\hat{\mathsf{k}}'$ such that $(\hat{\mathsf{k}}, (\hat{r}, \hat{\mathsf{k}}')) \in \mathcal{L}$ and $(\mathsf{ctxt}, \hat{\mathsf{k}}) = \mathsf{Encaps}(\mathsf{pk}, (\hat{r}, \hat{\mathsf{k}}))$	// G_6 - G_7

Figure A.1: The changes in the decapsulation oracle throughout the sequence of games.

Proof. We prove the Theorem via a sequence of games where changes of the specific games are shown to have at most only negligible advantage compared to the success probability in the BFKEM-IND-CCA security experiment. Let $\mathsf{Adv}_{\mathcal{A},j}$ be the advantage of $\mathcal{A}$ in Game j. Let Decaps' be the decryption oracle which we successively change (cf. Figure A.1 for the definition and all changes made throughout the sequence of games). The game steps are as follows:

Game G_0 (BFKEM-IND-CCA-security): Game 0 is the BFKEM-IND-CCA security experiment. Hence, we have that

$$\mathsf{Adv}_{\mathcal{A},0} = \mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cca}}_{\mathsf{BFKEM}',\mathcal{A}}(\lambda, m, k).$$

Game G_1 (γ-spreadness of ctxt): Game 1 is defined as Game 0 except that we substitute line 3 with line 4. More concretely, instead of checking $\mathsf{k} = \bot$, the decapsulation oracle checks if the adversary has queried G on (k) and maintains a list $\mathcal{L}$ with all adversarial queries to G as $(\mathsf{k}, \mathsf{G}(\mathsf{k})), \dots)$. The change is perfectly indistinguishable except for the case when the adversary inputs a ciphertext ctxt' such that $\mathsf{Decaps}'(\mathsf{sk}_i, \mathsf{ctxt}')$ behaves differently in Game 0 (i.e., $\mathsf{Decaps}(\mathsf{sk}_i, \mathsf{ctxt}') \neq \bot$) and Game 1 (i.e., $\mathsf{Decaps}(\mathsf{sk}_i, \mathsf{ctxt}') = \bot$). By the properties of BFKEM', we have that $(\mathsf{ctxt}, \mathsf{k}) = \mathsf{Encaps}(\mathsf{pk}; (r, \mathsf{k}))$ is determined by $(r, \mathsf{k}') = \mathsf{G}(\mathsf{k})$ for uniform $r \in \mathcal{R}$ and some $\mathsf{k} \in \mathcal{K}$. Hence, the different behavior can only happen if G was not queried before. But the probability that the adversary finds such ctxt' with $\mathsf{ctxt}' = \mathsf{ctxt}$ without querying $\mathsf{G}(\mathsf{k})$ is bounded by the γ-spreadness of BFKEM. Since the adversary queries the oracle at most $q_{\mathsf{G}} = q_{\mathsf{G}}(\lambda)$ times, we conclude $\mathsf{Adv}_{\mathcal{A},0} \leq \mathsf{Adv}_{\mathcal{A},1} + q_{\mathsf{G}} \cdot 2^{-\gamma}$.

Game G_2 (conceptional change): Game 2 is defined as Game 1 except that we substitute line 5 with line 6. More concretely, we read the unique tuple $(\mathsf{k}, (r, \mathsf{k}'))$ from the list $\mathcal{L}$ which guarantees that $(\mathsf{k}, (r, \mathsf{k}')) = (\mathsf{k}, \mathsf{G}(\mathsf{k}'))$ holds. Indeed, $\mathsf{G}(\mathsf{k})$ uniquely determines

$(\mathsf{k}, (r, \mathsf{k}'))$. We conclude $\mathsf{Adv}_{\mathcal{A},1} = \mathsf{Adv}_{\mathcal{A},2}$.

GAME G_3 (publicly-checkable puncturing of BFKEM): Game 3 is defined as Game 2 except that we introduce line 0. More concretely, we now first check if $\mathsf{CheckPunct}(\mathsf{pk}, \mathcal{L}', \mathsf{ctxt}) = \bot$, for some list of ciphertexts $\mathcal{L}'$. By the publicly-checkable puncturing property of BFKEM, we have that $\Pr[\mathsf{Decaps}(\mathsf{sk}_\ell, \mathsf{ctxt}) = \bot \iff \mathsf{CheckPunct}(\mathsf{pk}, \mathcal{L}', \mathsf{ctxt}) = \bot] \leq \delta$, for negligible error term $\delta = \delta(\lambda)$ and $\mathcal{L}' = (\mathsf{ctxt}_0, \ldots, \mathsf{ctxt}_{\ell-1})$ is the list of ciphertexts that were sent to Punc'. It follows that $\mathsf{Adv}_{\mathcal{A},2} \leq \mathsf{Adv}_{\mathcal{A},3} + \delta$.

GAME G_4 (extended-correctness of BFKEM): Game 4 is defined as Game 3 except that we substitute line 1 with line 2. More concretely, we now use the non-punctured (initial) secret key sk_0 to perform decryption of ctxt (note that sk_i can be an already punctured secret key). By the extended-correctness property of BFKEM, we have $\Pr[\mathsf{Decaps}(\mathsf{sk}_i, \mathsf{ctxt}) \neq \mathsf{Decaps}(\mathsf{sk}_0, \mathsf{ctxt})] \leq \delta$, for negligible error term $\delta = \delta(\lambda)$. Besides that, the oracle behaves the same as in Game 3. Hence, we conclude that $\mathsf{Adv}_{\mathcal{A},3} \leq \mathsf{Adv}_{\mathcal{A},4} + \delta$.

GAME G_5 (conceptional change): Game 5 is defined as Game 4 except that we simplify the checks in lines 4, 6, 7 and 8. More concretely, we simply replaced the checks in Game 4 with equivalent checks in Game 5 now in lines 9-10. Hence, we deduce $\mathsf{Adv}_{\mathcal{A},4} = \mathsf{Adv}_{\mathcal{A},5}$.

GAME G_6 (correctness for non-punctured secret keys of BFKEM): Game 6 is defined as Game 5 except that we check if there exist $(\hat{\mathsf{k}}, (\hat{r}, \hat{\mathsf{k}}')) \in \mathcal{L}$ such that $(\mathsf{ctxt}, \hat{\mathsf{k}}') = \mathsf{Encaps}(\mathsf{pk}; (\hat{r}, \hat{\mathsf{k}}))$ without comparing it to $\hat{\mathsf{k}} \leftarrow \mathsf{Decaps}(\mathsf{sk}_0, \mathsf{ctxt})$, that is we substitute lines 9-10 with lines 11-12. By the correctness for non-punctured secret keys of BFKEM, we have that if $(\mathsf{ctxt}, \hat{\mathsf{k}}) = \mathsf{Encaps}(\mathsf{pk}; (\hat{r}, \hat{\mathsf{k}}))$ then $\mathsf{Decaps}(\mathsf{sk}_0, \mathsf{ctxt}) = \hat{\mathsf{k}}$ except with negligible probability $\delta = \delta(\lambda)$. Hence, we infer that $\mathsf{Adv}_{\mathcal{A},5} \leq \mathsf{Adv}_{\mathcal{A},6} + \delta$.

GAME G_7 (conceptional change): Game 7 is defined as Game 6 except that we remove line 2 in Game 6. In Game 6, k' computed via $\mathsf{k} \leftarrow \mathsf{Decaps}(\mathsf{sk}_0, \mathsf{ctxt})$ was never used within the consistency checks anymore. Hence, we can safely remove this computation. We conclude $\mathsf{Adv}_{\mathcal{A},6} = \mathsf{Adv}_{\mathcal{A},7}$.

We are now ready to continue with the reduction to the BFKEM-IND-CPA-security of BFKEM. (In particular, note that in Game 7, sk_0 is not used anymore within the Decaps-oracle.) Let $\mathcal{A}$ be a PPT adversary on the BFKEM-IND-CCA-security of BFKEM', we will construct a PPT adversary $\mathcal{D}$ on the BFKEM-IND-CPA-security of BFKEM. $\mathcal{D}$ receives $(\mathsf{ctxt}^*, \mathsf{k}_b^*)$, for some (unknown) $b \leftarrow \{0,1\}$, that is forwarded to $\mathcal{A}$. During the experiment, oracle-calls by $\mathcal{A}$ to Punc' and Cor are re-directed to the BFKEM-IND-CPA-challenger. The decapsulation oracle Decaps' is as defined in Game 7. Eventually, $\mathcal{A}$ outputs a guess b' which $\mathcal{D}$ forwards to its challenger.

Analysis. We conclude that the success probability of $\mathcal{A}$ in the BFKEM-IND-CCA-security experiment is

$$\mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cca}}_{\mathsf{BFKEM}',\mathcal{A}}(\lambda, m, k) \leq \mathsf{Adv}^{\mathsf{bfkem\text{-}ind\text{-}cpa}}_{\mathsf{BFKEM},\mathcal{D}}(\lambda, m, k) + 3 \cdot \delta + \frac{q_{\mathsf{G}}}{2^\gamma}.$$

□

A.3 Evaluation

In this section, we present the evaluation of our compiler applied to all the NIST candidates with non-negligible correctness error. Throughout this section, $\mathsf{O}[\Pi, \ell]$ denotes either $\mathsf{T}^\star$ or $\mathsf{C}_{p,d}$ and the generic framework applied to Π with ℓ parallel ciphertexts. In the columns with the runtime, we present both the expected runtime of a parallelized implementation as well as a serial implementation of the Encaps and Decaps algorithms, i.e. p/s where p denotes the runtime of the parallel implementation and s denotes the runtime of the serial implementation. For the runtime of the Decaps algorithm, we assume that none of the underlying schemes returns $\perp$ on failure, i.e. we consider the worst case. We want to note that the target correctness error is not consistent, but all of them target $\leq 2^{-128}$ for all levels. Hence, we will target the same error. In case $\delta^{\ell-1}$ is only slightly larger than 2^{-128}, we also include it in the tables to give a more complete picture.

A.3.1 Code-based KEMs

Let's start with ROLLO. The designers specify two IND-CPA secure variants, namely ROLLO-I and ROLLO-III, with decoders having DFRs between 2^{-30} and 2^{-42}. Additionally, ROLLO-II is specified as IND-CCA secure variant with a negligible DFR of 2^{-128}.[2] While our transform does not render ROLLO-III more efficient than ROLLO-II, for ROLLO-I the picture is quiet different: while the ciphertexts of ROLLO-I combined with our transform are slightly larger than those of ROLLO-II, public key and ciphertext size combined is always smaller even if we overshoot the goal for the correctness error. Runtime-wise, a parallel implementation is faster, of course. For the L1 and L5 instances of ROLLO-I, the table also includes instances where our transform produces a correctness error that is only slightly larger than 2^{-128}. If the analysis of the decoder is improved only by a small amount, those instances would become the desired ones without overshooting the correctness error by too much. The full comparison is depicted in Table A.1.[3]

Next, we discuss BIKE. All parameter sets targeting an IND-CPA security are specified with a bit flipping decoder obtaining a DFR of $< 10^{-7} \approx 2^{-23.25}$. More in depth analysis of the decoder of BIKE estimates the actual DFR between 2^{-49} and 2^{-57} [SV19]. Hence we will base our comparison on a DFR of 2^{-49} and thus on the same δ-correctness since DFR coincides with δ-correctness for BIKE [DGKP21]. Sendrier and Vasseur also expect that by increasing the size of the underlying field by up to 15 %, the decoder would achieve a negligible DFR. For the IND-CCA secure version of BIKE, the backflip decoder [SV20] is used which achieves a negligible DFR. This decoder comes with the drawback, however, that at the time of the round 2 submission no constant-time implementation was available. A less efficient but constant-time version of the decoder

[2] In this section, we will base δ estimations on the DFR if not specified otherwise.

[3] Note that with the new parameters proposed in `https://groups.google.com/a/list.nist.gov/forum/#!topic/pqc-forum/p7o1N2-sXFw`, we can observe similar trade-offs.

was proposed recently [DGK20], though. For BIKE, our transform only improves the runtime in case the parallel implementation is used, though. As expected, the public key is smaller compared to the IND-CCA versions, yet the increase in the ciphertext outweighs the saving in the public key size. Overall, our transform applied to BIKE leads to a trade-off between runtime efficiency and size. The in-depth comparison is depicted in Tables A.2 to A.4.

Finally, we consider LEDAcrypt which directly starts from a deterministic PKE. Hence, we have to apply the direct product compiler with independent keys, but use our modified version $\mathsf{C}^{\star}_{\mathsf{p,d}}$. Its parameter sets are specified with a non-negligible DFR of 2^{-64} for the IND-CPA case and with negligible DFR for the IND-CCA case. With a DFR of 2^{-64}, the compiler ends up doubling the key and ciphertext sizes and end up with larger sum by 17% (for L5) to 38% (for L1). But in any case, the runtime figures for Encaps and Decaps significantly improve using a parallel implementation, resulting in a trade-off between bandwidth and runtime costs. See Table A.5 for the full comparison.

Table A.1: Sizes (in bytes) and runtimes (in ms) of ROLLO. Runtimes are taken from the optimized implementations.

KEM	CCA	δ	sk	pk	ctxt	$\sum$	KGen	Encaps	Decaps
ROLLO-I-L1	✗	2^{-30}	40	465	465	930	0.10	0.02	0.18
O[ROLLO-I-L1,4]	✓	2^{-116}	40	**465**	1860	**2325**	**0.10**	**0.02**/0.08	**0.24**/0.96
O[ROLLO-I-L1,5]	✓	$2^{-147.7}$	40	**465**	2325	**2790**	**0.10**	**0.02**/0.10	**0.26**/1.30
ROLLO-II-L1	✓	2^{-128}	40	1546	1674	3220	0.69	0.08	0.53
ROLLO-III-L1	✗	2^{-30}	40	634	1180	1814	0.03	0.04	0.14
O[ROLLO-III-L1,4]	✓	2^{-118}	40	**634**	4720	5354	**0.03**	**0.04**/0.16	**0.26**/1.04
O[ROLLO-III-L1,5]	✓	$2^{-147.7}$	40	**634**	5900	6534	**0.03**	**0.04**/0.20	**0.30**/1.50
ROLLO-I-L3	✗	2^{-32}	40	590	590	1180	0.13	0.02	0.36
O[ROLLO-I-L3,4]	✓	2^{-126}	40	**590**	2360	**2950**	**0.13**	**0.02**/**0.08**	**0.42**/1.68
ROLLO-II-L3	✓	2^{-128}	40	2020	2148	4168	0.83	0.09	0.69
ROLLO-III-L3	✗	2^{-36}	40	830	1580	2410	0.04	0.05	0.38
O[ROLLO-III-L3,4]	✓	2^{-142}	40	**830**	6320	7150	**0.04**	**0.05**/0.20	**0.53**/2.12
ROLLO-I-L5	✗	2^{-42}	40	947	1894	2841	0.20	0.03	0.69
O[ROLLO-I-L5,3]	✓	$2^{-124.4}$	40	**947**	5682	6629	**0.20**	**0.03**/**0.09**	**0.75**/2.25
O[ROLLO-I-L5,4]	✓	2^{-166}	40	**947**	7576	8523	**0.20**	**0.03**/0.12	**0.78**/3.12
ROLLO-II-L5	✓	2^{-128}	40	2493	2621	5114	0.79	0.10	0.84
ROLLO-III-L5	✗	2^{-42}	40	1138	2196	3334	0.05	0.07	0.63
O[ROLLO-III-L5,3]	✓	$2^{-124.4}$	40	**1138**	6588	7726	**0.05**	**0.07**/0.21	**0.77**/2.31
O[ROLLO-III-L5,4]	✓	2^{-166}	40	**1138**	8784	9922	**0.05**	**0.07**/0.28	0.84/3.36

Table A.2: Sizes and runtimes (millions of cycles) of BIKE L1. Runtimes are taken from the reference implementations.

KEM	CCA	δ	sk	pk	ctxt	$\sum$	KGen	Encaps	Decaps
BIKE-1-L1	✗	2^{-49}	1988	20326	20326	40652	0.21	0.24	3.13
O[BIKE-1-L1,3]	✓	$2^{-145.4}$	1988	**20326**	60978	81304	**0.21**	**0.24**/0.72	**3.61**/10.83
BIKE-1-CCA-L1	✓	2^{-128}	25546	23558	23558	47116	0.36	0.34	4.15
BIKE-2-L1	✗	2^{-49}	1988	10163	10163	20326	4.79	0.14	3.01
O[BIKE-2-L1,3]	✓	$2^{-145.4}$	1988	**10163**	30489	40652	**4.79**	**0.14**/0.42	**3.29**/9.88
BIKE-2-CCA-L1	✓	2^{-128}	25546	11779	12035	23814	6.32	0.20	4.12
BIKE-3-L1	✗	2^{-49}	1876	22054	22054	44108	0.17	0.24	3.95
O[BIKE-3-L1,3]	✓	$2^{-145.4}$	1876	**22054**	66162	88216	**0.17**	**0.24**/0.71	**4.42**/13.27
BIKE-3-CCA-L1	✓	2^{-128}	26414	24538	24794	49332	0.23	0.29	5.65
BIKE-BO3-L1	✗	2^{-49}	1876	11283	22054	33337	0.17	0.31	3.95
O[BIKE-BO3-L1,3]	✓	$2^{-145.4}$	1876	**11283**	66162	77445	**0.17**	**0.31**/0.92	**4.56**/13.68
BIKE-BO3-CCA-L1	✓	2^{-128}	26414	12525	24794	37319	0.28	0.35	5.65

Table A.3: Sizes and runtimes (millions of cycles) of BIKE L3. Runtimes are taken from the reference implementations.

KEM	CCA	δ	sk	pk	ctxt	$\sum$	KGen	Encaps	Decaps
BIKE-1-L3	✗	2^{-49}	3090	39706	39706	79412	0.40	0.44	8.33
O[BIKE-1-L3,3]	✓	$2^{-145.4}$	3090	**39706**	119118	158824	**0.40**	**0.44**/1.32	9.21/27.63
BIKE-1-CCA-L3	✓	2^{-128}	52732	49642	49642	99284	0.77	0.71	8.86
BIKE-2-L3	✗	2^{-49}	3090	19853	19853	39706	7.30	0.25	8.28
O[BIKE-2-L3,3]	✓	$2^{-145.4}$	3090	**19853**	59559	79412	**7.30**	**0.25**/0.75	8.79/26.36
BIKE-2-CCA-L3	✓	2^{-128}	52732	24821	25077	49898	9.89	0.39	8.57
BIKE-3-L3	✗	2^{-49}	2970	43366	43366	86732	0.34	0.46	9.01
O[BIKE-3-L3,3]	✓	$2^{-145.4}$	2970	**43366**	130098	173464	**0.34**	**0.46**/1.38	9.94/29.81
BIKE-3-CCA-L3	✓	2^{-128}	57056	54086	54342	108428	0.60	0.62	9.59
BIKE-BO3-L3	✗	2^{-49}	2970	21939	43366	65305	0.34	0.59	9.01
O[BIKE-BO3-L3,3]	✓	$2^{-145.4}$	2970	**21939**	130098	152037	**0.34**	**0.59**/1.76	10.18/30.55
BIKE-BO3-CCA-L3	✓	2^{-128}	57056	27299	54342	81641	0.61	0.75	9.59

Table A.4: Sizes and runtimes (millions of cycles) of BIKE L5. Runtimes are taken from the reference implementations.

KEM	CCA	δ	sk	pk	ctxt	$\sum$	KGen	Encaps	Decaps
BIKE-1-L5	✗	2^{-49}	4111	65498	65498	130996	0.72	0.79	20.05
O[BIKE-1-L5,3]	✓	$2^{-145.4}$	4111	**65498**	196494	261992	**0.72**	**0.79**/2.38	21.63/64.90
BIKE-1-CCA-L5	✓	2^{-128}	85578	81194	81194	162388	1.15	1.02	17.96
BIKE-2-L5	✗	2^{-49}	4110	32749	32749	65498	14.05	0.42	19.81
O[BIKE-2-L5,3]	✓	$2^{-145.4}$	4110	**32749**	98247	130996	**14.05**	**0.42**/1.25	20.64/61.91
BIKE-2-CCA-L5	✓	2^{-128}	85578	40597	40853	81450	16.95	0.57	17.63
BIKE-3-L5	✗	2^{-49}	4256	72262	72262	144524	0.55	0.75	21.00
O[BIKE-3-L5,3]	✓	$2^{-145.4}$	4256	**72262**	216786	289048	**0.55**	**0.75**/2.26	22.50/67.51
BIKE-3-CCA-L5	✓	2^{-128}	93990	89734	89990	179724	1.03	1.15	20.21
BIKE-BO3-L5	✗	2^{-49}	4256	36387	72262	108649	0.55	0.97	21.00
O[BIKE-BO3-L5,3]	✓	$2^{-145.4}$	4256	**36387**	216786	253173	**0.55**	**0.97**/2.92	22.94/68.82
BIKE-BO3-CCA-L5	✓	2^{-128}	93990	45123	89990	135113	1.07	1.41	20.21

Table A.5: Sizes (in bytes) and runtimes (in ms) of LEDAcrypt. The instances with postfix NN refer to those with non-negligible DFR. Runtimes are taken from the reference implementations.

KEM	CCA	δ	sk	pk	ctxt	$\sum$	KGen	Encaps	Decaps
LEDAcrypt-L1-NN	✗	2^{-64}	25	4488	4488	8976	0.29	0.13	0.42
O[LEDAcrypt-L1-NN,2]	✓	2^{-127}	50	8976	8976	17952	0.59	**0.13**/0.26	0.55/1.10
LEDAcrypt-L1	✓	2^{-128}	25	6520	6520	13040	0.55	0.16	0.55
LEDAcrypt-L3-NN	✗	2^{-64}	33	7240	7420	14660	0.91	0.26	0.91
O[LEDAcrypt-L3-NN,2]	✓	2^{-127}	66	14480	14840	29320	1.81	**0.26**/**0.52**	**1.17**/2.34
LEDAcrypt-L3	✓	2^{-128}	33	12032	12032	24064	1.53	0.54	1.25
LEDAcrypt-L5-NN	✗	2^{-64}	41	11136	11136	22272	2.52	0.14	1.41
O[LEDAcrypt-L5-NN,2]	✓	2^{-127}	82	22272	22272	44544	5.04	**0.14**/**0.29**	**1.55**/3.11
LEDAcrypt-L5	✓	2^{-128}	41	19040	19040	38080	4.25	0.84	2.28

A.3.2 Lattice-Based KEMs

The designers of ThreeBears [Ham19] specify both a IND-CPA secure version and an IND-CCA secure one for each security level they target: the parameters sets of the former achieve around 2^{-62} decryption error whereas those of the latter guarantee a decryption error $< 2^{-140}$. Such improvement is obtained by reducing the variance of the error distribution, while leaving all other parameters fixed, and therefore by incurring in a security loss. Our complier will thus double the ciphertext size in order to achieve negligible decryption error but keep the security level constant.

Next, we consider Round5 [GZB+19]. Its designers specify three different versions both for a CPA-secure KEM and for a CCA-secure PKE. Moreover each of them has three variants: two based on structured lattices (one using error-correcting codes and the other one not) and one based on unstructured ones. The transform, as expected, provides a smaller public keys this time too but the doubling ciphertext, as in the FrodoKEM case, outweighs this advantage: public key and ciphertext size combined is always at least thirty percent bigger when our transform is applied. The results are shown in Table A.6.

Finally, we also consider FrodoKEM [NAB+19]. While the NIST submission was specified with negligible correctness error, an earlier version of the scheme [BCD+16] was specified with non-negligible error. For the submission, the designers set parameters which achieve negligible decryption error (which in their case corresponds to decryption error less than $2^{-128}, 2^{-192}$ and 2^{-256} for target 1, 3 and 5 security level respectively). On the contrary, the earlier version of this scheme [BCD+16], that we denote by FrodoCCS, achieves only non-negligible failure probability. It is therefore possible to apply our transform to this primitive and compare its performance, in terms of ciphertext/public-key size and runtime, to its later versions. In this case, the only advantage of our transform is the public key size, which remains slightly smaller compared to the CCA versions. This comes the cost of a blow-up in the ciphertext size which exceeds significantly the aforementioned gain. The full comparison is depicted in Table A.7.

Table A.6: Sizes (in bytes) and runtimes (millions of cycles) of Round5. Runtimes of the PKEs are taken from the reference implementations and KEMs' ones are approximated starting from those of the CCA PKE used to construct them. A parameter set is denoted as R5N{1,D}-{1,3,5}-{KEM,PKE}{0,5}, where {1,D} refers whether it is a non-ring (1) or ring (D) parameter set, {1,3,5} refers to the NIST security level, and {0,5} identifies the number of correctable bits.

KEM	CCA	δ	sk	pk	ctxt	$\sum$	KGen	Encaps	Decaps
R5ND-1-PKE0d-cpa	✗	2^{-65}	128	634	682	1316	0.06	0.09	0.04
O[R5ND-1-PKE0d-cpa,2]	✓	2^{-129}	128	**634**	1364	1998	**0.06**	**0.09**/0.19	**0.14**/0.28
R5ND-1-KEM0d-cca	✓	2^{-155}	256	676	756	1432	0.07	0.10	0.14
R5ND-3-PKE0d-cpa	✗	2^{-71}	192	909	981	1890	0.14	0.21	0.11
O[R5ND-3-PKE0d-cpa,2]	✓	2^{-141}	192	**909**	1962	2871	0.14	0.21/0.42	0.33/0.65
R5ND-3-KEM0d-cca	✓	2^{-147}	384	983	1119	2102	0.09	0.14	0.19
R5ND-5-PKE0d-cpa	✗	2^{-64}	256	1178	1274	2452	0.16	0.25	0.13
O[R5ND-5-PKE0d-cpa,2]	✓	2^{-127}	256	**1178**	2548	3726	0.16	0.25/0.50	0.38/0.76
R5ND-5-KEM0d-cca	✓	2^{-143}	512	1349	1525	2874	0.10	0.17	0.24
R5N1-1-PKE0d-cpa	✗	2^{-66}	128	5214	5236	10450	2.77	4.05	0.19
O[R5N1-1-PKE0d-cpa,2]	✓	2^{-131}	128	**5214**	10472	15686	**2.77**	**4.05**/8.10	**4.24**/8.48
R5N1-1-KEM0d-cca	✓	2^{-146}	256	5740	5804	11544	3.52	5.31	5.42
R5N1-3-PKE0d-cpa	✗	2^{-65}	192	8834	8866	17700	6.69	10.10	0.28
O[R5N1-3-PKE0d-cpa,2]	✓	2^{-129}	192	**8834**	17732	26566	**6.69**	**10.10**/20.20	**10.38**/20.75
R5N1-3-KEM0d-cca	✓	2^{-144}	384	9660	9732	19392	6.78	10.20	10.60
R5N1-5-PKE0d-cpa	✗	2^{-77}	256	14264	14288	28552	14.00	18.60	0.81
O[R5N1-5-PKE0d-cpa,2]	✓	2^{-153}	256	**14264**	28576	42840	14.00	**18.60**/37.20	**19.41**/38.83
R5N1-5-KEM0d-cca	✓	2^{-144}	512	14636	14724	29360	12.70	19.20	19.60

Table A.7: Sizes (in bytes) and runtimes (millions of cycles) of FrodoKEM and FrodoCCS. Runtimes are taken from the reference implementations.

KEM	CCA	δ	sk	pk	ctxt	$\sum$	KGen	Encaps	Decaps
FrodoKEM-640-AES	✓	$2^{-138.7}$	10272	9616	9720	19336	1.38	1.86	1.75
FrodoKEM-976-AES	✓	$2^{-199.6}$	15664	15632	15744	31376	2.82	3.56	3.40
FrodoKEM-1344-AES	✓	$2^{-255.5}$	21568	21520	21632	43152	4.76	5.98	5.75
FrodoCCS-Classical†	✗	$2^{-36.2}$	7120	7104	7112	14216	0.00	0.00	0.00
O[FrodoCCS-Classical,4]	✓	$2^{-142.8}$	7120	7104	28448	35552	0.00	0.00/0.00	0.00/0.00
FrodoCCS-Recommended	✗	$2^{-38.9}$	11296	11280	11288	22568	2.94	3.48	0.34
O[FrodoCCS-Recommended,4]	✓	$2^{-153.6}$	11296	11280	45152	56432	2.94	3.48/13.94	10.79/43.16
FrodoCCS-Paranoid	✗	$2^{-33.8}$	12976	12960	12968	25928	3.25	4.26	0.39
O[FrodoCCS-Paranoid,4]	✓	$2^{-133.2}$	12976	12960	51872	64832	3.25	4.26/17.06	13.18/52.73

† No runtime numbers are available for this parameter set.

APPENDIX B

Appendix to Chapter 3

B.1 Additional Preliminaries

B.1.1 Adaptor Signatures

Next, we recall the formal definitions of adaptor signatures [AEE+21].

Definition B.1.1 (Adaptor Signatures)**.** *An adaptor signature scheme* Π_{AS} *w.r.t. a couple of hard relations* $R, \widetilde{R}$*, with* $R \subseteq \widetilde{R}$*, and a signature scheme* $\Pi_{\mathcal{DS}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *consists of algorithms* $(\mathsf{pSign}, \mathsf{pAdapt}, \mathsf{PreVerify}, \mathsf{Ext})$ *defined as:*

$\hat{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$*: The pre-sign algorithm takes as input a secret key* sk*, message* $m \in \{0,1\}^*$ *and statement* $Y \in L_R$*, outputs a pre-signature* $\hat{\sigma}$*.*

$0/1 \leftarrow \mathsf{PreVerify}(\mathsf{pk}, m, Y, \hat{\sigma})$*: The pre-verify algorithm takes as input a public key* pk*, message* $m \in \{0,1\}^*$*, statement* $Y \in L_R$ *and pre-signature* $\hat{\sigma}$*, outputs a bit* b*.*

$\sigma \leftarrow \mathsf{pAdapt}(\hat{\sigma}, y)$*: The adapt algorithm takes as input a pre-signature* $\hat{\sigma}$ *and witness* y*, outputs a signature* σ*.*

$y \leftarrow \mathsf{Ext}(\sigma, \hat{\sigma}, Y)$*: The extract algorithm takes as input a signature* σ*, pre-signature* $\hat{\sigma}$ *and statement* $Y \in L_R$*, outputs a witness* y *such that* $(Y, y) \in \widetilde{R}$*, or* $\perp$*.*

The correctness definition of adaptor signatures is described below.

Definition B.1.2 (Pre-signature Correctness)**.** *An adaptor signature scheme* Π_{AS} *satisfies pre-signature correctness if for every* $\lambda \in \mathbb{N}$*, every message* $m \in \{0,1\}^*$ *and every*

$\underline{\mathsf{aSigForge}_{\mathcal{A},\Pi_{\mathsf{AS}}}(\lambda)}$

$\mathcal{Q} := \emptyset (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$m \leftarrow \mathcal{A}^{\mathsf{Sign}\mathcal{O}(\cdot),\mathsf{pSign}\mathcal{O}(\cdot,\cdot)}(\mathsf{pk})$
$(Y, y) \leftarrow \mathsf{GenR}(1^\lambda)$
$\hat{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$
$\sigma \leftarrow \mathcal{A}^{\mathsf{Sign}\mathcal{O}(\cdot),\mathsf{pSign}\mathcal{O}(\cdot,\cdot)}(\hat{\sigma}, Y)$
return $(m \notin \mathcal{Q} \wedge \mathsf{Verify}(\mathsf{pk}, m, \sigma))$

$\underline{\mathsf{Sign}\mathcal{O}(m)}$

if $m \in \mathcal{Q}$
  return $\perp$
$\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
$\mathcal{Q} := \mathcal{Q} \cup \{m\}$
return σ

$\underline{\mathsf{pSign}\mathcal{O}(m, Y)}$

if $m \in \mathcal{Q}$
  return $\perp$
$\hat{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$
$\mathcal{Q} := \mathcal{Q} \cup \{m\}$
return $\hat{\sigma}$

Figure B.1: (Weak) Unforgeabiltiy experiment of adaptor signatures

statement/witness pair $(Y, y) \in R$*, the following holds:*

$$\Pr\left[\begin{array}{l} \mathsf{PreVerify}(\mathsf{pk}, m, Y, \hat{\sigma}) = 1 \\ \wedge\ \mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1 \\ \wedge\ (Y, y') \in \widetilde{R} \end{array} \middle| \begin{array}{l} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \hat{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y) \\ \sigma := \mathsf{pAdapt}(\hat{\sigma}, y) \\ y' := \mathsf{Ext}(\sigma, \hat{\sigma}, Y) \end{array} \right] = 1.$$

Next, we formally define the security properties of an adaptor signature scheme. We relax the definition of unforgeability, introduced in [AEE+21], by restricting the adversary to query any given message $m \in \{0,1\}^*$ only once to one of the two oracle, either $\mathsf{Sign}\mathcal{O}(\cdot)$ or $\mathsf{pSign}\mathcal{O}(\cdot,\cdot)$. Looking ahead, we require this relaxation in order to prove the security of our adaptor signature scheme. Our instantiation is based on the GPV signature scheme [GPV08], and it is proven secure in the random oracle model, by relying on the programmability of the RO. The above restriction allows us to apply the same technique to prove the security of the adaptor signature scheme, as the random oracle needs to be programmed at most once for any given message m. However, this relaxation does not seem to lead to any practical security consequence as in real-world application, typical signed messages contain a time-stamp, and thus users never get to sign the same message more than once. Moreover, one could rely on the probabilistic FDH version of the GVP signature in order to overcome such drawback: every time the pSign or Sign algorithms are executed on input a message m, a fresh salt t is sampled, the message $m||t$ is signed, and t is appended to the so produced signature. This modification is in fact equivalent to the introduced restriction of the adversary as the introduced salt forces the adversary, with high probability, to only get signatures of different messages (i.e., different $(m||t)$).

Definition B.1.3 (Weak Unforgeability)**.** *An adaptor signature scheme* Π_{AS} *is* aEUF-CMA *secure if for every PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\mathsf{negl}[]$ *such that:*

$$\Pr[\mathsf{aSigForge}_{\mathcal{A},\Pi_{\mathsf{AS}}}(\lambda) = 1] \leq \mathsf{negl}[\lambda]$$

where the experiment $\mathsf{aSigForge}_{\mathcal{A},\Pi_{\mathsf{AS}}}$ *is defined as follows:*

Definition B.1.4 (Weak Pre-signature Adaptability)**.** *An adaptor signature scheme* Π_{AS} *satisfies weak pre-signature adaptability if for any* $\lambda \in \mathbb{N}$*, any message* $m \in \{0,1\}^*$*,*

$\underline{\mathsf{aWitExt}_{\mathcal{A},\Pi_{\mathsf{AS}}}(\lambda)}$

$\mathcal{Q} := \emptyset$
$(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(m, Y) \leftarrow \mathcal{A}^{\mathsf{Sign}\mathcal{O}(\cdot),\mathsf{pSign}\mathcal{O}(\cdot,\cdot)}(\mathsf{pk})$
$\hat{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$
$\sigma \leftarrow \mathcal{A}^{\mathsf{Sign}\mathcal{O}(\cdot),\mathsf{pSign}\mathcal{O}(\cdot,\cdot)}(\hat{\sigma})$
$y' := \mathsf{Ext}(\mathsf{pk}, \sigma, \hat{\sigma}, Y)$
return $(m \notin \mathcal{Q} \wedge (Y, y') \notin \widetilde{R}$
$\wedge\, \mathsf{Verify}(\mathsf{pk}, m, \sigma))$

$\underline{\mathsf{Sign}\mathcal{O}(m)}$

if $m \in \mathcal{Q}$
  return $\perp$
$\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$
$\mathcal{Q} := \mathcal{Q} \cup \{m\}$
return σ

$\underline{\mathsf{pSign}\mathcal{O}(m, Y)}$

if $m \in \mathcal{Q}$
  return $\perp$
$\hat{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$
$\mathcal{Q} := \mathcal{Q} \cup \{m\}$
return $\hat{\sigma}$

Figure B.2: (Weak) Witness extractability experiment for adaptor signatures

any statement/witness pair $(Y, y) \in R$*, any key pair* $(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and any pre-signature* $\hat{\sigma} \leftarrow \{0,1\}^*$ *with* $\mathsf{PreVerify}(\mathsf{pk}, m, Y, \hat{\sigma}) = 1$ *we have:*

$$\Pr[\mathsf{Verify}(\mathsf{pk}, m, \mathsf{pAdapt}(\hat{\sigma}, y)) = 1] = 1$$

Definition B.1.5 (Weak Witness Extractability)**.** *An adaptor signature scheme* Π_{AS} *is* witness extractable *if for every PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}[]$ *such that the following holds:*

$$\Pr[\mathsf{aWitExt}_{\mathcal{A},\Pi_{\mathsf{AS}}}(\lambda) = 1] \leq \mathsf{negl}[\lambda]$$

where the experiment $\mathsf{aWitExt}_{\mathcal{A},\Pi_{\mathsf{AS}}}$ *is defined as follows*

B.1.2 Argument Systems

Definition B.1.6 (Hard Relation)**.** *For a relation* R*, with statement/witness* (Y, y)*, let* L_R *be the associated language defined as* $\{Y \mid \exists\, y \text{ s.t. } (Y, y) \in R\}$*. We say that* R *is a hard relation if the following holds:*

i) There exists a PPT sampling algorithm GenR *that on input* 1^λ *outputs a statement/witness pair* $(Y, y) \in R$*,*

ii) The relation is poly-time decidable,

iii) For all PPT $\mathcal{A}$ *the probability of* $\mathcal{A}$ *on input* Y *outputting a valid witness* y *is negligible.*

We recall the definition of a non-interactive zero-knowledge proof of knowledge (NIZK-PoK) with online extractors as introduced in [Fis05].

Definition B.1.7 (NIZK-PoK)**.** *A tuple* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *of PPT algorithms is called a NIZK with an online extractor for a relation* R*, and random oracle* $\mathcal{H}$*, if the following holds:*

i) Completeness: For all $\lambda \in \mathbb{N}$ and any $(Y, y) \in R$, it holds that

$$\mathsf{Verify}(\mathsf{pp}, Y, \mathsf{Prove}(\mathsf{pp}, Y, y)) = 1$$

except with negligible probability,

ii) Zero knowledge: If there exists a negligible function μ, a PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, such that for all $\lambda \in \mathbb{N}$, any $(Y, y) \in R$, and any PPT adversary $\mathcal{A}$, such that the following probability is bound by a negligible function μ.

$$\mathsf{Pr}\left[\begin{array}{l|l} b' = \mathcal{A}(\mathsf{pp}, Y, \pi) & b \leftarrow \{0,1\} \\ \wedge\ b = b' & \textit{If } b = 0 \\ & \quad \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ & \quad \pi \leftarrow \mathsf{Prove}(\mathsf{pp}, Y, y) \\ & \textit{else if } b = 1 \\ & \quad (\mathsf{pp}, \mathsf{st}_0) \leftarrow \mathcal{S}_1(1^\lambda) \\ & \quad \pi \leftarrow \mathcal{S}_2(\mathsf{pp}, \mathsf{st}_0, Y) \end{array}\right]$$

iii) Online Extractor: There exist a PPT online extractor K with access to the sequence of queries to the random oracle and its answers, such that given (Y, π), the algorithm K can extract the witness y with $(Y, y) \in R$.

B.2 GPV Adaptor Signatures

We consider hard languages of the form

$$\mathcal{L} := \{(\mathbf{A}, \mathbf{v}') \in \mathcal{R}_q^{\eta \times \ell} \times \mathcal{R}_q^{\eta} \mid \exists\ \mathbf{u}' \in \mathcal{R}^{\ell}\ s.t.\ , \mathbf{A} \cdot \mathbf{u}' = \mathbf{v}' \bmod q \wedge \|\mathbf{u}'\| \leq \beta^*\}.$$

We will consider the following hard relations R, $\widetilde{R}$, that capture witnesses used to adapt and extracted witnesses respectively, are given by

$$R_{\mathbf{A}} := \{(\mathbf{v}', \mathbf{u}') \in \mathcal{R}_q^{\eta} \times \mathcal{R}^{\ell} \mid \mathbf{v}' = \mathbf{A} \cdot \mathbf{u}' \bmod q \wedge \|\mathbf{u}'\| \leq \beta\},$$

and

$$\widetilde{R}_{\mathbf{A}} := \{(\mathbf{v}', \mathbf{u}') \in \mathcal{R}_q^{\eta} \times \mathcal{R}^{\ell} \mid \mathbf{v}' = \mathbf{A} \cdot \mathbf{u}' \bmod q \wedge \|\mathbf{u}'\| \leq \widetilde{\beta}\},$$

where $\beta \leq \widetilde{\beta}$. As done in Aumayr et. al. in [AEE+21], we slightly modify the hard relation for which the adaptor signature is defined in order to be able to extract the corresponding witness in the security experiments. Let $\Pi = (\Pi.\mathsf{Setup}, \Pi.\mathsf{Prove}, \Pi.\mathsf{Verify})$ be a NIZK-PoK with online extractor for the relation $R_{\mathbf{A}}$, as defined in Definition B.1.7. We will consider the relation $R_{\mathbf{A}}^+$, whose statements are pairs $(\mathbf{v}', \pi)$, where $(\mathbf{v}', \mathbf{u}') \in R_{\mathbf{A}}$, and $\pi \leftarrow \Pi.\mathsf{Prove}(\mathsf{pp}, \mathbf{v}', \mathbf{u}')$, for $\mathsf{pp} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$. That is

$$R_{\mathbf{A}}^+ := \{((\mathbf{v}', \pi), \mathbf{u}') \mid \mathbf{v}' = \mathbf{A} \cdot \mathbf{u}' \bmod q \wedge \|\mathbf{u}'\| \leq \beta \wedge \Pi.\mathsf{Verify}(\mathsf{pp}, \mathbf{v}', \pi) = 1\}.$$

$\mathsf{Setup}(1^\lambda)$

$\mathbf{A} \leftarrow \mathcal{R}_q^{\eta \times \ell}$
$\tilde{\mathsf{pp}} \leftarrow \Pi.\mathsf{Setup}(1^\lambda)$
return $\mathsf{pp} := (\mathbf{A}, \tilde{\mathsf{pp}})$

$\mathsf{KeyGen}(\mathsf{pp})$

$\mathbf{X}^\intercal \leftarrow (\mathsf{SampD}(1^\eta, 1^\ell, \mathcal{R}, d))^\ell$
$\mathsf{sk} := \mathbf{X}$
$\mathsf{pk} := \mathbf{Y} := \mathbf{A} \cdot \mathbf{X} \bmod q$
return $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Sign}(\mathsf{sk}, m \in \mathcal{M})$

$\mathbf{u} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \rho}$
$\mathbf{v} := \mathbf{A} \cdot \mathbf{u} \bmod q$
$\mathbf{c} := \mathbf{G}^{-1}(\mathbf{v} - H(m))$
$\mathbf{z} := \mathbf{u} + \mathbf{X} \cdot \mathbf{c}$
return $\sigma := (\mathbf{c}, \mathbf{z})$

$\mathsf{Verify}(\mathsf{pk}, m, \sigma)$

return $\begin{cases} \mathbf{A} \cdot \mathbf{z} - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} \stackrel{?}{=} H(m) \bmod q \\ \|(\mathbf{c}, \mathbf{z})\| \leq \gamma_2 \end{cases}$

$\mathsf{pSign}(\mathsf{sk}, m, Y = (\mathbf{v}', \pi))$

if $\Pi.\mathsf{Verify}(\tilde{\mathsf{pp}}, \mathbf{v}', \pi) = 0$
  return $\perp$
$\mathbf{u} \leftarrow \mathcal{D}_{\mathcal{R}^\ell, \rho}$
$\mathbf{v} := \mathbf{A} \cdot \mathbf{u} \bmod q$
$\mathbf{c} := \mathbf{G}^{-1}(\mathbf{v} + \mathbf{v}' - H(m))$
$\hat{\mathbf{z}} := \mathbf{u} + \mathbf{X} \cdot \mathbf{c}$
return $\hat{\sigma} := (\mathbf{c}, \hat{\mathbf{z}})$

$\mathsf{PreVerify}(\mathsf{pk}, m, \mathbf{v}', \hat{\sigma})$

return $\begin{cases} \mathbf{A} \cdot \hat{\mathbf{z}} - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} \stackrel{?}{=} H(m) - \mathbf{v}' \bmod q \\ \|(\mathbf{c}, \hat{\mathbf{z}})\| \leq \gamma_1 \end{cases}$

$\mathsf{pAdapt}(\hat{\sigma}, \mathbf{u}')$

$\mathbf{z} := \hat{\mathbf{z}} + \mathbf{u}'$
return $\sigma := (\mathbf{c}, \mathbf{z})$

$\mathsf{Ext}(\sigma, \hat{\sigma}, \mathbf{v}')$

return $\mathbf{u}' := \mathbf{z} - \hat{\mathbf{z}}$

Figure B.3: GPV based adaptor signatures using a NIZK-PoK Π.

Since $R_{\mathbf{A}}$ is a hard relation, so is $R_{\mathbf{A}}^+$. In order to ease readability and avoid introducing too many different notations, in our construction we replace $R_{\mathbf{A}}$ with $R_{\mathbf{A}}^+$.

Parameters. The scheme parameters

- $\rho \geq (d \cdot \ell \cdot \sqrt{\ell} + \beta) \cdot \sqrt{Q}$, where Q is the maximum number of oracle queries allowed in the experiment,
- β, witness norm bound,
- $\gamma_1 \geq \rho \cdot \sqrt{\ell}$, norm bound for pre-signature,
- $\gamma_2 \geq \gamma_1 + \beta$, norm bound for signature,

- $\widetilde{\beta} \geq \gamma_1 + \gamma_2$, norm bound of extracted witnesses,

have to be chosen so that $M\text{-}\mathsf{SIS}_{\mathcal{R}_q,\eta,\ell,2\gamma_2+2d\ell\sqrt{\ell}}$ and $M\text{-}\mathsf{SIS}_{\mathcal{R}_q,\eta,\ell,\gamma_1+\gamma_2+\beta+2d\ell\sqrt{\ell}}$ are hard.

B.2.1 Security Analysis

Pre-signature correctness follows via a straightforward investigation, using the fact that $\mathbf{A} \cdot \mathbf{u}' = \mathbf{v}' \bmod q$.

Lemma B.2.1 (Weak Pre-signature Adaptability)**.** *The adaptor signature scheme described in Fig. B.3 satisfies weak pre-signature adaptability with respect to the relation* $R_{\mathbf{A}}$.

Proof. Let $\hat{\sigma} = (\mathbf{c}, \hat{\mathbf{z}})$ be a valid pre-signature with $\mathsf{PreVerify}(\mathsf{pk}, \mathbf{m}, \mathbf{v}', \hat{\sigma}) = 1$, and $\mathbf{u}'$, with $\|\mathbf{u}'\| \leq \beta$ be a witness corresponding to $\mathbf{v}'$. Since $\hat{\sigma}$ is valid, we have $\|\hat{\mathbf{z}}\| \leq \gamma_1$. Then, $\mathsf{pAdapt}(\hat{\sigma}, \mathbf{u}') = (\mathbf{c}, \hat{\mathbf{z}} + \mathbf{u}') = \sigma$. Therefore, we have

$$\|\mathbf{z}\| = \|\hat{\mathbf{z}} + \mathbf{u}'\| \leq \|\hat{\mathbf{z}}\| + \|\mathbf{u}'\| \leq \gamma_1 + \beta \leq \gamma_2.$$

We further have

$$\begin{aligned} \mathbf{A} \cdot \mathbf{z} - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} &= \mathbf{A} \cdot (\hat{\mathbf{z}} + \mathbf{u}') - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} \\ &= (\mathbf{A} \cdot \hat{\mathbf{z}} - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c}) + \mathbf{A} \cdot \mathbf{u}' \\ &= (H(m) - \mathbf{v}') + \mathbf{v}' = H(m) \bmod q. \end{aligned}$$

From the above two equations, it follows that σ is a valid signature for message $\mathbf{m}$, i.e., $\mathsf{Verify}(\mathsf{pk}, \mathbf{m}, \sigma) = 1$. □

Lemma B.2.2 (Weak Unforgeability)**.** *Let* $\Pi = (\Pi.\mathsf{Setup}, \Pi.\mathsf{Prove}, \Pi.\mathsf{Verify})$*, used in the construction on the adaptor signature from Fig. B.3, be a NIZK-PoK with online extractor for the relation* $R_{\mathbf{A}}$*. Assuming* $M\text{-}\mathsf{SIS}_{\mathcal{R}_q,\eta,\ell,2\gamma_2+2d\ell}$ *and that* $\rho \geq (d\ell + \beta)\sqrt{Q}$*, where* Q *is the maximum number of oracle queries an attacker can make, the adaptor signature from Fig. B.3 is weakly unforgeable in the random oracle model.*

Proof. We prove the unforgeability of the adaptor signature scheme by reduction to the M-SIS problem. Let $(\mathbf{A}, \mathbf{v}^*)$ be the given M-SIS instance. Consider the following sequence of hybrids. In all of them let $\sigma^* = (\mathbf{c}^*, \mathbf{z}^*)$ be the forgery signature output by $\mathcal{A}$ on message m^*. Without loss of generality, we can assume that the adversary always queries the random oracle H on every message m before making a presigning/signing query on m.

– Hybrid Hyb_0: This is identical to the real experiment.

– Hybrid Hyb_1: This is identical to the real experiment except that the public parameters $\tilde{\mathsf{pp}}$ of the NIZK-PoK are generated by the simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, whose existence is guaranteed by the zero-knowledge property of the NIZK-PoK Π (Definition B.1.7), i.e., $(\tilde{\mathsf{pp}}, \mathsf{state}_0) \leftarrow \mathcal{S}_1(1^\lambda)$. Moreover whenever the adversary outputs a challenge message m^*, the challenger samples $(\mathbf{v}^*, \mathbf{u}^*) \leftarrow \mathsf{GenR}(1^\lambda)$, runs the simulator S on input $(\tilde{\mathsf{pp}}, \mathsf{state}_0, \mathbf{v}^*)$, to obtain a simulated proof π^*, i.e., $\pi^* \leftarrow \mathsf{S}(\tilde{\mathsf{pp}}, \mathsf{state}_0, \mathbf{v}^*)$, and returns $(\hat{\sigma}, (\mathbf{v}^*, \pi^*))$ to the adversary.

– Hybrid Hyb_2: Here the simulator $\mathcal{S}$ works as follows:

  – The simulator records a list $\mathcal{Q}$ of all H queries made by $\mathcal{A}$ with their responses. Let $Q = |\mathcal{Q}|$ be the number of hash queries made by $\mathcal{A}$.

  – Whenever $\mathcal{A}$ queries the random oracle H on input m, the simulator samples $\mathbf{v} \leftarrow \mathcal{R}_q^\ell$, $\mathbf{z} \leftarrow \mathcal{D}_{\mathcal{R}^\ell,\rho}$, sets $\mathbf{c} := \mathbf{G}^{-1}(\mathbf{v})$, programs the random oracle $H(m) := \mathbf{A} \cdot \mathbf{z} - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} \bmod q$, and returns $\sigma = (\mathbf{c}, \mathbf{z})$, stores $(m, H(m), \sigma)$ in $\mathcal{Q}$, and returns $H(m)$ to the adversary.

  – Whenever the adversary queries the $\mathsf{Sign}\mathcal{O}(\cdot)$ oracle on input m, the simulator finds the corresponding entry $(m, H(m), \sigma)$ in $\mathcal{Q}$, and returns σ to the adversary.

  – Whenever the adversary queries the $\mathsf{pSign}\mathcal{O}(\cdot)$ oracle on input $(m, (\mathbf{v}', \pi))$, the simulator checks the validity of $(\mathbf{v}', \pi))$, extracts the witness $\mathbf{u}'$ of $\mathbf{v}'$ from the proof π, finds the corresponding entry $(m, H(m), \sigma = (\mathbf{c}, \mathbf{z}))$ in $\mathcal{Q}$, and returns $\hat{\sigma} = (\mathbf{c}, \hat{\mathbf{z}} := (\mathbf{z} - \mathbf{u}'))$ to the adversary.

  – Whenever the adversary outputs a challenge message m^*, the simulator finds the corresponding entry $(m^*, H(m^*), \sigma = (\mathbf{c}, \mathbf{z}))$ in $\mathcal{Q}$, runs $(\mathbf{v}^*, \mathbf{u}^*) \leftarrow \mathsf{GenR}(1^\lambda)$, and returns $(\hat{\sigma} := (\mathbf{c}, \hat{\mathbf{z}} := \mathbf{z} - \mathbf{u}^*), (\mathbf{v}^*, \pi^*))$ to the adversary, where π^* is the corresponding simulated NIZK-PoK proof.

- Hybrid Hyb_3: This is identical to hybrid Hyb_2, except that this time the simulator samples $\mathbf{X}^\intercal \leftarrow (\mathsf{SampD}(1^\eta, 1^\ell, \mathcal{R}, d))^\ell$, and sets $\mathbf{Y} := \mathbf{A} \cdot \mathbf{X} - \mathbf{G} \bmod q$.

Let δ_i denote the probability of an adversary winning in hybrid Hyb_i. Hybrids Hyb_0 and Hyb_1 only differ in the way the proof π^* is generated. By the zero-knowledge property of the proof system NIZK-PoK Π, one has that the distribution of simulated proofs is computationally indistinguishable to the distribution of real ones. Therefore, we obtain

$$\delta_0 \leq \delta_1 + \mathsf{negl}(\lambda).$$

Claim B.2.1. *If there is an adversary that makes at most Q oracle queries and can win the game in hybrid Hyb_1 with probability δ_1, then its probability of winning in hybrid Hyb_2 is polynomial in δ_1, if $\rho \geq (d \cdot \ell + \beta) \cdot \sqrt{Q}$.*

Proof. The only difference between the two hybrids is in the value of $\mathbf{z}$ or $\hat{\mathbf{z}}$. For $i \in [Q]$, in hybrid Hyb_1 we have $\mathbf{z}_i$ or $\hat{\mathbf{z}}_i$ equal to $\mathbf{u}_i + \mathbf{X} \cdot \mathbf{c}_i$ with $\mathbf{u}_i \leftarrow \mathcal{D}_{\mathcal{R}^\ell,\rho}$, while in hybrid

Hyb_2, we have $\mathbf{z}_i \leftarrow \mathcal{D}_{\mathcal{R}^\ell,\rho}$ and $\hat{\mathbf{z}}_i \leftarrow \mathcal{D}_{\mathcal{R}^\ell,\rho,-\mathbf{u}'}$. Let us refer to the joint distribution of all $\mathbf{z}$ and $\hat{\mathbf{z}}$ in Hyb_1 as D_1 and that in Hyb_2 as D_2. Let E denote the event that the adversary wins the game. Then, by our assumptions, we have $D_1(E) = \delta_1$. From the probability preservation property (Lemma 3.2.1) of the Rényi divergence, we get

$$D_2(E) \geq \frac{\delta_1^{\frac{a}{a-1}}}{R_a(D_1||D_2)}, \quad \text{for any } a \in (1,\infty).$$

In order to compute $R_a(D_1||D_2)$, notice that, for $i \in [Q]$, the vectors $\mathbf{z}_i$ or $\hat{\mathbf{z}}$ are drawn from distribution $D_{1i} = \mathcal{D}_{\mathcal{R}^\ell,\rho,\mathbf{X}\cdot\mathbf{c}_i}$ in hybrid Hyb_1, and from distribution $D_{2i} = \mathcal{D}_{\mathcal{R}^\ell,\rho}$ or $D_{2i} = \mathcal{D}_{\mathcal{R}^\ell,\rho,-\mathbf{u}'}$ in hybrid Hyb_2. Notice that $D_1 = (D_{11},\ldots,D_{1Q})$, and $D_2 = (D_{21},\ldots,D_{2Q})$. By Lemma 3.2.2, we have

$$R_a(D_{1i}||D_{2i}) \leq \exp\left(a\cdot\pi\cdot\frac{(\|\mathbf{X}\cdot\mathbf{c}_i\|+\|\mathbf{u}'\|)^2}{\rho^2}\right), \quad \text{for any } a \in (1,\infty).$$

Since each row of $\mathbf{X}$ has norm bounded by d, and $\|\mathbf{c}_i\| \leq \sqrt{\ell}$, we have $\|\mathbf{X}\cdot\mathbf{c}_i\| \leq d\ell$. Moreover, the extracted witness must have $\|\mathbf{u}'\| \leq \beta$ as the NIZK-PoK proof π verifies correctly. Using the multiplicativity property of the Rényi divergence (Lemma 3.2.1), we get

$$R_a(D_1||D_2) \leq \exp\left(a\cdot\pi\cdot\frac{Q\cdot(d\cdot\ell+\beta)^2}{\rho^2}\right).$$

Using the assumption $\rho \geq (d\cdot\ell+\beta)\cdot\sqrt{Q}$, we get that $R_a(D_1||D_2) \leq \exp(a\cdot\pi)$. Therefore, we obtain that $\delta_2 := D_2(E) \geq \delta_1^{\frac{a}{a-1}}/\exp(a\cdot\pi)$. Taking any value of $a > 1$ yields the result. □

Hybrids Hyb_2 and Hyb_3 only differ in the way the public key $\mathbf{Y}$ is generated. By the properties of SampD, we have that $\mathbf{A}\cdot\mathbf{X} \bmod q$ is statistically close to uniform. Thus, the same holds for $\mathbf{A}\cdot\mathbf{X}-\mathbf{G} \bmod q$, which implies that

$$\delta_2 \leq \delta_3 + \mathsf{negl}(\lambda).$$

Claim B.2.2. *If there is an adversary $\mathcal{A}$ that makes at most Q oracle queries, and succeeds in forging a valid signature with probability δ_3 is hybrid Hyb_3, then we can define an algorithm $\mathcal{B}$ which given $\mathbf{A} \leftarrow \mathcal{R}_q^{\eta\times\ell}$, finds a non-zero short $\mathbf{u} \in \mathcal{R}^\ell$ such that $\|\mathbf{u}\| \leq 2\gamma_2 + 2d\ell$ and $\mathbf{A}\cdot\mathbf{u} = \mathbf{0} \mod q$.*

Proof. Let $\sigma^* = (\mathbf{c}^*,\mathbf{z}^*)$ be the forgery signature output by $\mathcal{A}$ on message m^*, $\mathbf{v}^*$ the challenge statement provided to $\mathcal{A}$, and $\sigma = (\mathbf{c},\mathbf{z})$ the corresponding signature created when the adversary queried the random oracle on message m^*. Let $\mathbf{u} := \mathbf{z}^*-\mathbf{z}-\mathbf{X}(\mathbf{c}^*-\mathbf{c})$. We have

$$\begin{aligned}\mathbf{A}\cdot(\mathbf{z}^*-\mathbf{z}+\mathbf{X}\cdot(\mathbf{c}^*-\mathbf{c})) &= \mathbf{A}\cdot(\mathbf{z}^*-\mathbf{z}) - (\mathbf{G}+\mathbf{Y})\cdot(\mathbf{c}^*-\mathbf{c})\\ &= H(m) - H(m)\\ &= \mathbf{0} \bmod q.\end{aligned}$$

Moreover, since $\|\mathbf{z}^*\| \leq \gamma_2$, $\|\mathbf{z}\| \leq \gamma_2$, and $\|\mathbf{X} \cdot \mathbf{c}^*\|, \|\mathbf{X} \cdot \mathbf{c}\| \leq d\ell$, we obtain $\|\mathbf{u}\| \leq 2\gamma_2 + 2d\ell$. It remain to argue that $\mathbf{u} \neq \mathbf{0}$. We distinguish 2 cases:

Case 1: $\mathbf{c}^* = \mathbf{c}$. In this case we have $\mathbf{A} \cdot (\mathbf{z}^* - \mathbf{z}) = \mathbf{0}$. Recall that presignature given to $\mathcal{A}$ corresponding to statement $\mathbf{v}^*$ was of the form $\hat{\sigma} = (\mathbf{c}, \mathbf{z} - \mathbf{u}^*)$. We obtain

$$\begin{aligned}\mathbf{A} \cdot (\mathbf{z}^* - \mathbf{z}) &= \mathbf{A} \cdot (\mathbf{z}^* - (\hat{\mathbf{z}} + \mathbf{u}^*)) \\ &= \mathbf{A} \cdot ((\mathbf{z}^* - \hat{\mathbf{z}}) - \mathbf{u}^*) \\ &= \mathbf{0} \bmod q,\end{aligned}$$

which implies that $\mathbf{A} \cdot (\mathbf{z}^* - \hat{\mathbf{z}}) = \mathbf{v}^* = \mathbf{A} \cdot \mathbf{u}^* \bmod q$. As argued by Gentry et. al in [GPV08], the min-entropy of $\mathbf{u}^*$ given $\mathbf{v}^*$ (and also $\hat{\mathbf{z}}$ in our case) is $\omega(\log k)$. Thus, $\mathbf{z}^* - \hat{\mathbf{z}} \neq \mathbf{u}^*$, except with negligible probability.

Case 2: $\mathbf{c}^* \neq \mathbf{c}$. In this case, we can apply the same arguments as in Lemma 5.4 of [Lyu12], to get that $\mathbf{u} \neq \mathbf{0}$ with high probability. This proves the claim. □

Thus, by showing that $\delta_3 \leq \mathsf{negl}(\lambda)$, this finishes the proof. □

Lemma B.2.3 (Witness Extractability)**.** *Let* $\Pi = (\Pi.\mathsf{Setup}, \Pi.\mathsf{Prove}, \Pi.\mathsf{Verify})$*, used in the construction on the adaptor signature from Fig. B.3, be a NIZK-PoK with online extractor for the relation* $R_{\mathbf{A}}$*. Assuming* $M\text{-}\mathsf{SIS}_{\mathcal{R}_q,\eta,\ell,\gamma_1+\gamma_2+\beta+2d\ell}$ *and that* $\rho \geq (d\ell+\beta)\sqrt{Q}$*, where* Q *is the maximum number of oracle queries an attacker can make, the adaptor signature from Fig. B.3 is witness extractable in the random oracle model.*

Proof. We prove the witness extractibility of the adaptor signature scheme by reduction to the M-SIS problem. Let $\mathbf{A}$ be the given M-SIS instance. The proof is very similar to that of Lemma B.2.2. Consider the following sequence of hybrids. In all of them let $\sigma^* = (\mathbf{c}^*, \mathbf{z}^*)$ be the forgery signature output by $\mathcal{A}$ on message m^*, and let $(\mathbf{v}^*, \pi^*)$ be the challenge statement. Without loss of generality, we can assume that the adversary always queries the random oracle H on every message m before making a presigning/signing query on m.

- Hybrid Hyb_0: This is identical to the real experiment.
- Hybrid Hyb_1: Here the challenger works as follows:
  - The simulator records a list $\mathcal{Q}$ of all H queries made by $\mathcal{A}$ with their responses. Let $Q = |\mathcal{Q}|$ be the number of hash queries made by $\mathcal{A}$.
  - Whenever $\mathcal{A}$ queries the random oracle H on input m, the simulator samples $\mathbf{v} \leftarrow \mathcal{R}_q^\ell$, $\mathbf{z} \leftarrow \mathcal{D}_{\mathcal{R}^\ell,\rho}$, sets $\mathbf{c} := \mathbf{G}^{-1}(\mathbf{v})$, programs the random oracle $H(m) := \mathbf{A} \cdot \mathbf{z} - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} \bmod q$, and returns $\sigma = (\mathbf{c}, \mathbf{z})$, stores $(m, H(m), \sigma)$ in $\mathcal{Q}$, and returns $H(m)$ to the adversary.

– Whenever the adversary queries the $\mathsf{Sign}\mathcal{O}(\cdot)$ oracle on input m, the simulator finds the corresponding entry $(m, H(m), \sigma)$ in $\mathcal{Q}$, and returns σ to the adversary.

– Whenever the adversary queries the $\mathsf{pSign}\mathcal{O}(\cdot)$ oracle on input $(m, (\mathbf{v}', \pi))$, the simulator checks the validity of $(\mathbf{v}', \pi))$, extracts the witness $\mathbf{u}'$ of $\mathbf{v}'$ from the proof π, finds the corresponding entry $(m, H(m), \sigma = (\mathbf{c}, \mathbf{z}))$ in $\mathcal{Q}$, and returns $\hat{\sigma} = (\mathbf{c}, \hat{\mathbf{z}} := (\mathbf{z} - \mathbf{u}'))$ to the adversary.

– Whenever the adversary outputs a challenge message-statement tuple $(m^*, (\mathbf{v}^*, \pi^*))$, the simulator works as if it was responding to a presignature query: it makes use of the extractor K, whose existence is guaranteed by the extractability property of the NIZK-PoK Π (Definition B.1.7). In order to extract the witness $\mathbf{u}^*$ corresponding to the statement $\mathbf{v}^*$, it runs extractor K, with access to the random oracle and its answers, on input $(\mathbf{v}^*, \pi^*)$, i.e., $\mathbf{u}^* \leftarrow \mathsf{K}(\mathbf{v}^*, \pi^*)$. Then, it finds the entry $(m^*, H(m^*), \sigma = (\mathbf{c}, \mathbf{z}))$ in $\mathcal{Q}$ corresponding to m^*, and returns $\hat{\sigma} = (\mathbf{c}, \hat{\mathbf{z}} := (\mathbf{z} - \mathbf{u}^*))$ to the adversary.

- Hybrid Hyb_2: This is identical to hybrid Hyb_1, except that this time the simulator samples $\mathbf{X}^{\mathsf{T}} \leftarrow (\mathsf{SampD}(1^\eta, 1^\ell, \mathcal{R}, d))^\ell$, and sets $\mathbf{Y} := \mathbf{A} \cdot \mathbf{X} - \mathbf{G} \bmod q$.

Let δ_i denote the probability of an adversary winning in hybrid Hyb_i.

Claim B.2.3. *If there is an adversary that makes at most Q oracle queries and can win the game in hybrid Hyb_0 with probability δ_0, then its probability of winning in hybrid Hyb_1 is polynomial in δ_0, if $\rho \geq (d \cdot \ell + \beta)\sqrt{Q}$.*

Proof. The proof is identical to that of the analogous claim used in the proof of Lemma B.2.2. □

Hybrids Hyb_1 and Hyb_2 only differ in the way the public key $\mathbf{Y}$ is generated. By the properties of SampD, we have that $\mathbf{A} \cdot \mathbf{X} \bmod q$ is statistically close to uniform. Thus, the same holds for $\mathbf{A} \cdot \mathbf{X} - \mathbf{G} \bmod q$, which implies that

$$\delta_1 \leq \delta_2 + \mathsf{negl}(\lambda).$$

Claim B.2.4. *If there is an adversary $\mathcal{A}$ that makes at most Q oracle queries, and succeeds winning with probability δ_2 in hybrid Hyb_2, then we can define an algorithm $\mathcal{B}$ which given $\mathbf{A} \leftarrow \mathcal{R}_q^{\eta \times \ell}$, finds a non-zero short $\mathbf{u}^*$ such that $\|\mathbf{u}^*\| \leq \gamma_1 + \gamma_2 + \beta + 2 \cdot d \cdot \ell$ and $\mathbf{A} \cdot \mathbf{u}^* = 0 \bmod q$.*

Proof. Let $\sigma^* = (\mathbf{c}^*, \mathbf{z}^*)$ be the forged signature output by $\mathcal{A}$. We distinguish 2 cases:

Case 1: $\mathbf{c}^* = \mathbf{c}$. Since both pre-signature and signature verify, we have that

$$\mathbf{A} \cdot \mathbf{z}^* - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} = H(m) \bmod q \qquad \text{and} \qquad \mathbf{A} \cdot \hat{\mathbf{z}} - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} = H(m) - \mathbf{v}^* \bmod q,$$

from which we obtain that

$$\mathbf{A} \cdot (\mathbf{z}^* - \hat{\mathbf{z}}) = \mathbf{v}^* \bmod q.$$

As $\|\mathbf{z}^* - \hat{\mathbf{z}}\| \leq \gamma_1 + \gamma_2 \leq \widetilde{\beta}$, the output of the Ext algorithm $\mathbf{u}^* := \mathbf{z}^* - \hat{\mathbf{z}}$ is a valid witness for $\mathbf{v}^*$.

Case 2: $\mathbf{c}^* \neq \mathbf{c}$. In this case, we make use of the extractability property of the zero-knowledge proof π^*, in order to extract $\mathbf{u}^*$ and obtain from the forged signature a M-SIS solution. Let $\mathbf{u}^* \leftarrow \mathsf{K}(\mathbf{v}, \pi, \mathcal{H})$, where $\mathcal{H}$ is the list of random oracle queries made by $\mathcal{A}$. With high probability, it holds that $((\mathbf{v}^*, \pi^*), \mathbf{u}^*) \in R_{\mathbf{A}}$. Using that

$$\mathbf{A} \cdot \mathbf{z}^* - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c}^* = H(m) \bmod q \qquad \text{and} \qquad \mathbf{A} \cdot \hat{\mathbf{z}} - (\mathbf{G} + \mathbf{Y}) \cdot \mathbf{c} = H(m) - \mathbf{v}^* \bmod q,$$

we obtain

$$[\mathbf{A} | \mathbf{A} \cdot \mathbf{X}] \cdot \begin{bmatrix} \mathbf{z}^* - \hat{\mathbf{z}} + \mathbf{u}^* \\ \mathbf{c}^* - \mathbf{c} \end{bmatrix} = 0 \bmod q,$$

which leads to the non-zero M-SIS solution $\mathbf{r} := \mathbf{z}^* - \hat{\mathbf{z}} + \mathbf{u}^* + \mathbf{X} \cdot (\mathbf{c}^* - \mathbf{c})$, with $\|\mathbf{r}\| \leq \gamma_1 + \gamma_2 + \beta + 2 \cdot d \cdot \ell$, by relying again on the analysis done in [Lyu12, Lemma 5.4]. □

Putting everything together, this concludes the proof. □

B.3 On Achieving (Functional) Hiding

We discuss potential approaches to modify the VC construction in Section 3.5 to achieve hiding and functional hiding.

Definition B.3.1 ((Functional) Hiding)**.** *A VC scheme for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ is said to be statistically/computationally hiding if for any $\lambda, w, t \in \mathbb{N}$, any $\mathsf{pp} \in \mathsf{Setup}(1^\lambda, 1^w, 1^t)$, and any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^w$, the distributions*

$$\{\, c : (c, \mathsf{aux}) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x}) \,\} \qquad \textit{and} \qquad \{\, c : (c, \mathsf{aux}) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x}') \,\}$$

are statistically/computationally indistinguishable.

A VC scheme for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ is said to be statistically/computationally functional hiding if there exists a tuple of PPT simulators $\mathcal{S} = (\mathcal{S}_0, \mathcal{S}_1)$ such that, for any $\lambda, w, t \in \mathbb{N}$ and any $(f, \mathbf{x}, y) \in \mathcal{F}_{w,t} \times \mathcal{X}^w \times \mathcal{Y}^t$ satisfying $f(\mathbf{x}) = y$, the distributions

$$\left\{ \begin{array}{r} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^w, 1^t) \\ (\mathsf{pp}, c, \pi) : (c, \mathsf{aux}) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x}) \\ \pi \leftarrow \mathsf{Open}(\mathsf{pp}, f, \mathsf{aux}) \end{array} \right\} \qquad \textit{and} \qquad \left\{ (\mathsf{pp}, c, \pi) : \begin{array}{l} (\mathsf{pp}, \mathsf{td}) \leftarrow \mathcal{S}_0(1^\lambda, 1^w, 1^t) \\ (c, \pi) \leftarrow \mathcal{S}_1(\mathsf{td}, f, y) \end{array} \right\}$$

are statistically/computationally indistinguishable.

In the VC construction in Figure 3.1, a commitment of $\mathbf{x}$ is of the form $\langle \mathbf{v}, \mathbf{x} \rangle \bmod q$ as in essentially every lattice-based commitment schemes. A well-known technique for achieving hiding is to commit instead to the concatenation of $\mathbf{x}$ and a short random vector $\mathbf{r}$. If the randomness vector $\mathbf{r}$ has sufficiently many dimensions one could argue that $\langle \mathbf{v}, (\mathbf{x}, \mathbf{r}) \rangle \bmod q$ is statistically close to uniform. This can be done, relying on the regularity lemmas discussed in Section 4.3.2. Achieving functional hiding requires more work. In the following, we discuss three (potential) approaches on top of introducing $\mathbf{r}$.

Notice that the verification algorithm in Figure 3.1 is simply checking that an opening proof $(\mathbf{u}_0, \mathbf{u}_1)$ satisfies two SIS relations. An approach of achieving functional hiding is therefore to replace the opening proof $(\mathbf{u}_0, \mathbf{u}_1)$ with a zero-knowledge proof of knowledge of $(\mathbf{u}_0, \mathbf{u}_1)$. This can be done efficiently using Schnorr-like proofs in the random oracle model, without affecting compactness since the witness $(\mathbf{u}_0, \mathbf{u}_1)$ and the relation that it satisfies are of size independent of (f, y). Due to the use of a random oracle, the resulting scheme may no longer be purely algebraic (depending on how the random oracle is heuristically instantiated) and therefore might not be recursively composed natively. However, in applications where a single party performs the entire recursive composition, it is possible to first recursively compose the non-functional-hiding scheme in Figure 3.1, and finish off with a zero-knowledge proof of the final opening proof.

Another approach, related to the first and inspired by [CLMQ21], is to (provably) instantiate the random oracle in a Schnorr-like proof with a function that outputs short preimages of the inputs with respect to a linear function. While this technique preserves the algebraic structure of the scheme, it requires each of the witness components $\mathbf{u}_0$ and $\mathbf{u}_1$ to be a short square matrix instead of a short vector. In other words, to achieve functional hiding using this approach, we need to either introduce dummy relations or prove ℓ openings in batch.

The third approach is to argue directly that $(\mathbf{u}_0, \mathbf{u}_1)$ leaks no information about $\mathbf{x}$. This is intuitively plausible since both $\mathbf{u}_0$ and $\mathbf{u}_1$ consists of linear combinations of Gaussian vectors with coefficients depending on $\mathbf{r}$. Indeed, for $d = 1$, we could apply a Gaussian-version of the Leftover Hash Lemma [AGHS13] and rejection sampling to argue this formally. For $d \geq 2$, unfortunately, the distributions of $\mathbf{u}_0$ and $\mathbf{u}_1$ become much more complicated, making generalising the argument for $d = 1$ to $d \geq 2$ difficult. Furthermore, we remark that this approach relies on making the variance of $\mathbf{u}_0$ and $\mathbf{u}_1$ super-polynomially wide to "smudge" the contribution of $\mathbf{x}$. This means the modulus q would also need to be super-polynomially large.

B.4 Vector Commitments without Knowledge Assumptions

We strip off components for compactness and extractability from our main VC construction in Section 3.5. The resulting scheme supports the same class of openings. It achieves the

weaker notions of succinctness and weak binding but does not rely on any non-falsifiable assumption.

B.4.1 Definitions

Since our goal is to achieve succinctness, we fix $t = 1$ everywhere and omit it from the syntax. The definition of correctness is modified accordingly. Next, we formalise (weak) binding and succinctness.

Definition B.4.1 ((Weak) Binding)**.** *Let $\rho : \mathbb{N}^3 \to [0,1]$. A VC scheme for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ is said to be weakly ρ-binding if for any pair of PPT adversary $\mathcal{A}$ and any $s, w \in \mathsf{poly}(\lambda)$ it holds that the following expression is upper-bounded by $\rho(\lambda, s, w)$:*

$$\Pr\left[\begin{array}{l} \forall\, i \in \{0,1\}, \\ \quad \mathsf{Verify}(\mathsf{pp}_{f_i,y_i}, \mathbf{z}_i, c, \pi_i) = 1, \\ \wedge\ f_0(\mathbf{z}_0, \cdot) = f_1(\mathbf{z}_1, \cdot) \\ \wedge\ y_0(\mathbf{z}_0) \neq y_1(\mathbf{z}_1) \end{array} \middle| \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w) \\ \left(c, (f_i, \mathbf{z}_i, y_i, \pi_i)_{i=0}^1\right) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \forall\, i \in \{0,1\}, \\ \quad \mathsf{pp}_{f_i,y_i} \leftarrow \mathsf{PreVerify}(\mathsf{pp}, (f_i, y_i)) \end{array}\right].$$

We say that the scheme is weakly binding if it is weakly ρ-binding and $\rho(\lambda, s, w)$ is negligible in λ for any $s, w \in \mathsf{poly}(\lambda)$.

The scheme is said to be ρ-binding if for any PPT adversary $\mathcal{A}$ and $w, t = \mathsf{poly}(\lambda)$ it holds that the following expression is upper-bounded by $\rho(\lambda)$:

$$\Pr\left[\begin{array}{l} \left(\forall\, i \in I,\ \mathsf{Verify}(\mathsf{pp}_{f_i,y_i}, \mathbf{z}_i, c, \pi_i) = 1\right) \\ \wedge\ \neg\left(\exists\, \mathbf{x} \in \mathcal{K}^w,\ \forall\, i \in I,\ f_i(\mathbf{z}_i, \mathbf{x}) = y_i(\mathbf{z}_i)\right) \end{array} \middle| \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w) \\ \left(c, I, (f_i, \mathbf{z}_i, y_i, \pi_i)_{i \in \mathbb{Z}_t}\right) \leftarrow \mathcal{A}(\mathsf{pp}) \\ \forall\, i \in I \\ \quad \mathsf{pp}_{f_i,y_i} \leftarrow \mathsf{PreVerify}(\mathsf{pp}, (f_i, y_i)) \end{array}\right].$$

We say that the scheme is binding if it is ρ-binding and $\rho(\lambda, s, w)$ is negligible in λ for any $s, w \in \mathsf{poly}(\lambda)$.

Note that in the binding definition the existence of $\mathbf{x}$ is checked over the base field $\mathcal{K}$ rather than the ring $\mathcal{R}$. The reason for this choice will become clear when we discuss the binding property of our construction.

For positional openings [CF13] weak binding and binding are trivially equivalent. Using linear algebra, it is also not difficult to that the equivalence also holds for openings to linear functions over finite fields [LRY16, LM19].[1] The equivalence does not seem to hold, however, for openings to linear functions over rings nor for high-degree openings over rings or fields.

[1] In [LM19], the generic group model is used to prove the binding property of the compact linear map commitment construction. If the compactness requirement is dropped, binding could be proven in the plain model.

Definition B.4.2 (Succinctness)**.** *A VC scheme for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ is said to be succinct if there exists $p(\lambda, s, w) \in \mathsf{poly}(\lambda, \log s, \log w)$ such that for any $\lambda, s, w \in \mathbb{N}$, any $\mathsf{pp} \in \mathsf{Setup}(1^\lambda, 1^s, 1^w)$, any $(f, \mathbf{z}, \mathbf{x}, y) \in \mathcal{F}_{s,w} \times \mathcal{X}^s \times \mathcal{X}^w \times \mathcal{Y}_s$, any $(c, \mathsf{aux}) \in \mathsf{Com}(\mathsf{pp}, \mathbf{x})$, and any $\pi \in \mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$, it holds that $|c| \leq p(\lambda, s, w)$ and $|\pi| \leq p(\lambda, s, w)$, where $|\cdot|$ denotes the description size.*

B.4.2 Construction

$\mathsf{Setup}(1^\lambda, 1^s, 1^w)$

$\mathbf{v} \leftarrow (\mathcal{R}_q^\times)^w$

$(\mathbf{A}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\eta, 1^\ell, q, \mathcal{R}, \beta)$

$\mathbf{t} \leftarrow \mathcal{R}_q^\eta$

$\mathbf{u}_g \leftarrow \mathsf{SampPre}(\mathsf{td}, g(\mathbf{v}) \cdot \mathbf{t}, \beta), \ \forall g \in \mathcal{G}$

return $\mathsf{pp} := \left(\mathbf{A}, \mathbf{t}, (\mathbf{u}_g)_{g \in \mathcal{G}}, \mathbf{v}\right)$

$\mathsf{Com}(\mathsf{pp}, \mathbf{x})$

$c := \langle \mathbf{v}, \mathbf{x} \rangle \bmod q$

for $\mathbf{e} \in \bigcup_{k \in [d]} \mathcal{E}_k$ **do**

$$\mathbf{u}_\mathbf{e} := d! \cdot \sum_{\mathbf{e}' \in \mathcal{E}_k \setminus \{\mathbf{e}\}} \frac{\binom{k}{\mathbf{e}'}}{\binom{k}{\mathbf{e}}} \cdot \mathbf{x}^{\mathbf{e}'} \cdot \mathbf{u}_{\mathbf{X}^{\mathbf{e}'-\mathbf{e}}}$$

$\mathsf{aux} := (\mathbf{u}_\mathbf{e})_{\mathbf{e} \in \bigcup_{k \in [d]} \mathcal{E}_k}$

return (c, aux)

$\mathsf{PreVerify}(\mathsf{pp}, (f, y))$

if $(f, y) \notin \mathcal{F}_{s,w} \times \mathcal{Y}_s$ **then return** $\perp$

$$\hat{f}(\mathbf{Z}, C) := d! \cdot \left(\sum_{k=1}^{d} \sum_{\mathbf{e} \in \mathcal{E}_k} \binom{k}{\mathbf{e}}^{-1} \cdot f_\mathbf{e}(\mathbf{Z}) \cdot \mathbf{v}^{-\mathbf{e}} \cdot C^k - y(\mathbf{Z}) \right)$$

$\mathsf{pp}_{f,y} := \left(\mathbf{A}, \mathbf{t}, \hat{f}\right)$

return $\mathsf{pp}_{f,y}$

$\mathsf{Open}(\mathsf{pp}, f, \mathbf{z}, \mathsf{aux})$

$$\mathbf{u} := \sum_{k=1}^{d} \sum_{\mathbf{e} \in \mathcal{E}_k} f_\mathbf{e}(\mathbf{z}) \cdot \mathbf{u}_\mathbf{e}$$

return $\pi := \mathbf{u}$

$\mathsf{Verify}(\mathsf{pp}_{f,y}, \mathbf{z}, c, \pi)$

$b_0 := \left(\mathbf{A}\mathbf{u} \stackrel{?}{=} \hat{f}(\mathbf{z}, c) \cdot \mathbf{t} \bmod q\right)$

$b_1 := \left(\|\mathbf{u}\| \stackrel{?}{\leq} \delta\right)$

return $b_0 \wedge b_1$

Figure B.4: Stripped-Down VC Construction.

A formal description of the stripped-down construction is in Figure B.4. The proof of

correctness is completely analogous to that of Theorem 3.5.1 and is therefore omitted.

Theorem B.4.3. *For $d = O(1)$, $\ell \geq \mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$ and*

$$\delta = 3 \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{2d+1} \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2},$$

the VC construction in Figure B.4 is correct.

Theorem B.4.4. *The construction of vector commitments for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ in Figure B.4 is weakly-binding if $\ell \geq \mathsf{lhl}(\mathcal{R}, \eta, q, \beta)$, $\beta \geq \alpha$, and the k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta,\ell,w,\mathcal{G},1,\mathcal{D},\mathcal{T},\beta,2\delta}$ assumption holds, where $\mathcal{D}$ is such that the distribution*

$$\left\{ (\mathbf{A}, \mathbf{t}, \{\, \mathbf{u}_{\mathcal{G}} \,\}, \mathbf{v}) \middle| \begin{array}{l} \mathbf{A} \leftarrow \mathcal{R}_q^{\eta \times \ell};\ \mathbf{t} \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^{\times})^w \\ \mathbf{u}_g \leftarrow \mathcal{D}_{g,\mathbf{A},\mathbf{t},\mathbf{v}},\ \forall g \in \mathcal{G} \end{array} \right\}$$

is statistically close to the distribution

$$\left\{ (\mathbf{A}, \mathbf{t}, \{\, \mathbf{u}_{\mathcal{G}} \,\}, \mathbf{v}) \middle| \begin{array}{l} \mathbf{A} \leftarrow \mathcal{R}_q^{\eta \times \ell};\ \mathbf{t} \leftarrow \mathcal{T};\ \mathbf{v} \leftarrow (\mathcal{R}_q^{\times})^w \\ \mathbf{u}_g \leftarrow \mathsf{SampD}(1^{\eta_i}, 1^{\ell_i}, \mathcal{R}, \beta) : \mathbf{A} \cdot \mathbf{u}_g \equiv g(\mathbf{v}) \cdot \mathbf{t} \bmod q,\ \forall g \in \mathcal{G} \end{array} \right\}.$$

Proof. Let $\mathcal{A}$ be an adversary against the weakly binding property of the construction in Figure B.4. We construct an algorithm $\mathcal{B}$ for the k-M-$\mathsf{ISIS}_{\mathcal{R}_q,\eta,\ell,w,\mathcal{G},1,\mathcal{D},\mathcal{T},\beta,2\delta}$ problem. Our algorithm $\mathcal{B}$ inputs a problem instance $\left(\mathbf{A}, \mathbf{t}, \mathbf{v}, \{\, \mathbf{u}_g \,\}_{g \in \mathcal{G}}\right)$, sets $\mathsf{pp} := \left(\mathbf{A}, \mathbf{t}, \mathbf{v}, \{\, \mathbf{u}_g \,\}_{g \in \mathcal{G}}\right)$, and runs $\mathcal{A}$ on pp to obtain a tuple $(c, (f_i, \mathbf{z}_i, y_i, \mathbf{u}_i)_{i=0}^1)$. Our algorithm $\mathcal{B}$ outputs $(s^*, \mathbf{u}_{g^*}) = (d! \cdot (y_1(\mathbf{z}_1) - y_0(\mathbf{z}_0)), \mathbf{u}_0 - \mathbf{u}_1)$.

Suppose $\mathcal{A}$ is a successful adversary against the weak-binding property of our VC construction. By our assumption on $\mathcal{D}$, the distribution of the public parameters pp passed to $\mathcal{A}$ by $\mathcal{B}$ is statistically close to that generated by Setup. Therefore, with non-negligible probability, the tuple that $\mathcal{A}$ returns to $\mathcal{B}$ satisfies

$$\begin{cases} \mathbf{A} \cdot \mathbf{u}_i = \hat{f}_i(\mathbf{z}_i, c) \cdot \mathbf{t} \bmod q, \\ \|\mathbf{u}_i\| \leq \delta. \end{cases}$$

for $i \in \{0, 1\}$ with $f_0(\mathbf{z}_0, \cdot) = f_1(\mathbf{z}_1, \cdot)$ but $y_1(\mathbf{z}_1) \neq y_0(\mathbf{z}_0)$, which implies $\mathbf{A} \cdot \mathbf{u}_{g^*} = s^* \cdot \mathbf{t} \bmod q$, $0 < \|s^*\| \leq 2\,\delta$, and $\|\mathbf{u}_{g^*}\| \leq 2\,\delta$. □

Theorem B.4.5. *For $n \in \mathsf{poly}(\lambda)$, $q, \delta \in \mathsf{poly}(\lambda, s, w)$, and $\ell \in \Theta(\log q) = \mathsf{polylog}(\lambda, s, w)$, covering the choices of parameters in Theorems B.4.3 and B.4.4, the VC construction in Figure B.4 is succinct.*

Concretely, let $\mathcal{R}$ be a power-of-2 cyclotomic ring so that $\gamma = n$. For $s = w \geq n$ and for the following choices of parameters,

$$\begin{aligned} & d = O(1), \quad \beta \geq \alpha, \\ & \delta = 3 \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{2d+1} \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2}, \\ & q \approx \delta \cdot n \cdot \log n, \ \textit{and} \\ & \ell = \mathsf{lhl}(\mathcal{R}, \eta, q, \beta) \approx 2 \log_\beta q, \end{aligned}$$

a commitment is of size $O(n \log s)$*, and an opening proof is of size* $O(n \cdot (\log s + \log \beta)^2 / \log \beta)$*. The minimum is attained at* $\beta = \Theta(s)$*, where an opening proof is of size* $O(n \log s)$*.*

Proof. For the general case, we observe that a commitment $c \in \mathcal{R}_q$ is of description size $n \log q \in \mathsf{poly}(\lambda, \log s, \log w)$, and an opening proof $\mathbf{u}$ is of description size $n \cdot \ell \cdot \log \delta \in \mathsf{poly}(\lambda, \log s, \log w)$.

For the concrete case, we have

$$\begin{aligned}
\delta &= 3 \cdot (s+d)^d \cdot (w+d)^{2d} \cdot \alpha^{2d+1} \cdot \beta \cdot \gamma_{\mathcal{R}}^{2d+2} = O(s^{3d} \cdot \alpha^{2d+1} \cdot \beta \cdot n^{2d+2}), \\
q &= \delta \cdot n \cdot \log n = O(s^{3d} \cdot \alpha^{2d+1} \cdot \beta \cdot n^{2d+3} \cdot \log n), \\
\log \delta, \log q &= O(\log s + \log \beta), \\
\ell &= 2 \log q / \log \beta = O((\log s + \log \beta)/ \log \beta), \\
|c| &= n \cdot \log q = O(n \log s), \text{ and} \\
|\mathbf{u}| &= n \cdot \ell \cdot \log \delta \\
&= n \cdot O((\log s + \log \beta)/ \log \beta) \cdot O(\log s + \log \beta) \\
&= O(n \cdot (\log s + \log \beta)^2 / \log \beta).
\end{aligned}$$

□

B.4.3 On Binding

We study to what extent binding can be achieved without relying on non-falsifiable assumptions.

In the following informal discussion we omit the public input $\mathbf{z}$ for readability. As mentioned previously, in the case where $\mathcal{F}$ consists of only position maps, then weak binding is trivially equivalent to binding. This is because, if f_i are position maps, i.e. $f_i(\mathbf{x}) = x_i$, for $i \in I$ then the only way to force that no $\mathbf{x} \in \mathcal{K}^w$ satisfies $f_i(\mathbf{x}) = y_i$ for all $i \in I$ is to set $f_{i'} = f_{i''}$ but $y_{i'} \neq y_{i''}$ for some distinct $i', i'' \in I$.

In fact, even if $\mathcal{F}$ consists of only linear maps, i.e. $d = 1$, the equivalence between weak binding and binding still holds without considering the norm bound constraint, e.g. when the linear maps are defined over a finite field. Indeed, suppose that $f_i(\mathbf{x}) = y_i$ for all $i \in I$ is not satisfiable by any $\mathbf{x} \in \mathcal{K}^w$, then by Gaussian elimination one can find a coefficient vector $\mathbf{r}' \in \mathcal{K}^I$ such that $\sum_{i \in I} r'_i f_i \equiv 0$ and $\sum_{i \in I} r'_i y_i = 1$. Multiplying $\mathbf{r}'$ by the least common multiple Δ of the denominators in $\mathbf{r}'$ to obtain $\mathbf{r} \in \mathcal{R}^I$, we have $\sum_{i \in I} r_i f_i \equiv 0$ and $\sum_{i \in I} r_i y_i = \Delta$. Since the verification algorithm Verify is linear in (f, y), we obtain openings for (f_i, y_i) and $(f_i, y_i + \Delta)$.

In the lattice setting, however, we need to argue that Δ is not too large relative to (a large enough) q, so that we can use the technique in the proof of Theorem B.4.4 to turn an adversary against binding into an algorithm for solving certain k-M-ISIS problems.

The following theorem states that binding can be achieved for $d = 1$ and an exponentially large q.

Theorem B.4.6. *In addition to the assumptions made in Theorem B.4.4, let* $d = 1$, $\delta^* := |I| \cdot \gamma_{\mathcal{R}} \cdot \delta \cdot \nu^{\nu} \cdot \alpha^{2\nu}$, *and* $q \geq \omega\left(|I| \cdot \gamma_{\mathcal{R}} \cdot \delta \cdot \nu^{\nu} \cdot \alpha^{2\nu}\right)$ *where* $\nu := (w+1) \cdot n$. *If the VC construction in Figure B.4 is weakly-binding for* δ^* *then it is also binding for* δ.

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ against binding, we construct a PPT adversary $\mathcal{B}$ against weak binding as follows. Our adversary $\mathcal{B}$ receives the public parameters $\left(\mathbf{A}, \mathbf{t}, \mathbf{v}, \{\, \mathbf{u}_g \,\}_{g \in \mathcal{G}}\right)$ and forwards it to $\mathcal{A}$. By assumption, $\mathcal{A}$ outputs a tuple $(c, I, \{\, f_i, \mathbf{z}_i, y_i, \mathbf{u}_i \,\}_{i \in I})$ which satisfies the following with non-negligible probability:

1. For all $i \in I$, $\mathbf{A} \cdot \mathbf{u}_i \equiv \hat{f}_i(\mathbf{z}_i, c) \bmod q$.
2. For all $i \in I$, $\|\mathbf{u}_i\| \leq \delta$.
3. There does not exist $\mathbf{x} \in \mathcal{K}^w$ such that, for all $i \in I$, $f_i(\mathbf{z}_i, \mathbf{x}) = y_i$.

Since f_i is a homogeneous linear polynomial, we have $\hat{f}_i(\mathbf{z}_i, \cdot) = f_i(\mathbf{z}_i, \cdot) - y(\mathbf{z}_i)$ and $f_i(\mathbf{z}_i, \cdot)$ can be represented by a vector $\mathbf{f}_i \in \mathcal{R}^w$ such that $f_i(\mathbf{z}_i, \mathbf{x}) = \langle \mathbf{f}_i, \mathbf{x} \rangle$ for any $\mathbf{x} \in \mathcal{K}^w$. Let $\mathbf{F}$ be the matrix with the i-th column being $\mathbf{f}_i$, $\mathbf{U}$ be the matrix with the i-th column being $\mathbf{u}_i$, and $\mathbf{y} = (y_i(\mathbf{z}_i))_{i \in I}$. Consequently, we can rewrite the equations in Item 1 above as

$$\mathbf{A}^{\mathsf{T}} \cdot \mathbf{U} \equiv \mathbf{t} \cdot \begin{pmatrix} c \cdot \bar{\mathbf{v}}^{\mathsf{T}} & -1 \end{pmatrix} \begin{pmatrix} \mathbf{F} \\ \mathbf{y}^{\mathsf{T}} \end{pmatrix} \bmod q.$$

By assumption there exists an $\mathbf{r}' \in \mathcal{K}^{|I|}$ s.t. $\begin{pmatrix} \mathbf{F} \\ \mathbf{y}^{\mathsf{T}} \end{pmatrix} \cdot \mathbf{r}' = (0, \ldots, 0, 1)$. Thus, we have $\mathbf{r} := \Delta \cdot \mathbf{r}' \in \mathcal{R}^{|I|}$ s.t. $\begin{pmatrix} \mathbf{F} \\ \mathbf{y}^{\mathsf{T}} \end{pmatrix} \cdot \mathbf{r} = (0, \ldots, 0, \Delta)$ where Δ is the least common multiple of the denominators in $\mathbf{r}'$. Note that a solution in $\mathcal{R}$ maps to a solution over $\mathbb{Z}$ by the map $g \mapsto \mathrm{rot}(g)$. To bound $\|\mathbf{r}\|$ and Δ, assume $|I| = w + 1$, which represents the worst case, and apply known bounds for solutions over $\mathbb{Z}$ [MS04, Fact 25]: $\Delta \leq \nu^{\nu/2} \cdot \alpha^{\nu}$ and $\|\mathbf{r}\| \leq \nu^{\nu/2} \cdot \alpha^{\nu-1} \cdot \Delta$.

Let $\mathbf{u}'_0 := \mathbf{U} \cdot \mathbf{r}$. We have

$$\begin{aligned} \mathsf{Verify}(\mathsf{pp}_{f_0, y_0}, \mathbf{z}_0, c, \mathbf{u}_0) &= 1 \\ \mathsf{Verify}(\mathsf{pp}_{f_0, y_0 + \Delta}, \mathbf{z}_0, c, \mathbf{u}'_0) &= 1 \\ \|\mathbf{u}_0\| < \|\mathbf{u}_{|I|}\| &\leq \delta^* \end{aligned}$$

but $y_0 \neq y_0 + \Delta$. □

We next discuss why proving binding in the case $d > 1$ from falsifiable assumptions seems unlikely. Indeed, if we were given a compact and binding VC for degree-d openings for $d \geq 2$, we can construct a SNARG for the NP-complete language of degree-d polynomial maps satisfiability (Section 3.6), where a SNARG is almost a SNARK but only satisfies soundness instead of knowledge soundness. Due to the impossibility result of Gentry and Wichs [GW11], who showed that certain flavour of SNARG requires non-falsifiable assumption or non-black-box reduction, we obtain the same impossibility for compact and binding VC with openings to non-linear polynomial maps.

B.4.4 On Compactness

We discuss the difficulty of achieving compactness without relying on the knowledge k-M-ISIS assumption.

For VC constructions where the verification equation is linear in the opening proof, such as the constructions presented in Section 3.5.1, and Appendix B.4.2, a natural strategy to achieve compactness is to aggregate multiple opening proofs using a random linear combination. Instantiating the strategy involves deciding how the random coefficients of the linear combination are generated.

For schemes where the verification equation is defined over prime-order cyclic groups, provably binding ways of instantiating the strategy includes (i) embedding the random coefficients in the public parameters and prove soundness in the generic [LM19] or algebraic [GRWZ20] group model, (ii) making the verification interactive and let the verifier sample the coefficients, or (iii) generate the coefficients using a random oracle. The proofs of binding in all three approaches rely crucially on the fact that Vandermonde matrices defined by distinct elements in a finite field are always invertible.

In the lattice setting, the random coefficients need to be chosen from a subtractive set, i.e. a set where the difference between any pairs of distinct elements is always invertible, for a similar proof strategy to work (see, e.g. [AL21]). Unfortunately, it has been shown [AL21] that over many cyclotomic rings $\mathcal{R}$, the size of (even relaxed variants of) subtractive sets is at most $O(n)$, which is insufficient for aggregating an unbounded polynomial number t of opening proofs into a single proof of size poly-logarithmic in t.

B.4.5 Post-Quantum Security

We analyse the security of our stripped-down construction against quantum attackers. We show that our construction, viewed as an ordinary commitment scheme, satisfies the notion of collapsing [Unr16]. This is done is two steps: First, we show that our VC scheme satisfies the notion of somewhere statistically binding (SSB) [HW15]. Next, we rely on a previous result, reproduced in Appendix B.4.5 that an SSB VC is also collapsing.

$\mathsf{KeyGen}(1^\lambda)$	$\mathsf{Enc}(\mathsf{pk}, m \in \mathcal{R}_p)$
$f' \leftarrow \mathsf{SampD}(1^1, 1^1, \mathcal{R}, \beta); f \coloneqq p \cdot f' + 1$	$(s, e) \leftarrow \mathsf{SampD}(1^1, 1^2, \mathcal{R}, \beta')$
if $f \notin \mathcal{R}_q^\times$ resample	**return** $c \coloneqq h \cdot s + p \cdot e + m$
$g \leftarrow \mathsf{SampD}(1^1, 1^1, \mathcal{R}, \beta)$	$\mathsf{Dec}(\mathsf{sk}, c)$
if $g \notin \mathcal{R}_q^\times$ resample	**return** $m \coloneqq (f \cdot c \bmod q) \bmod p$
return $\mathsf{pk} \coloneqq h = p \cdot g/f$, $\mathsf{sk} \coloneqq f$	

Figure B.5: NTRU Encryption. n, q, p, β, β' are parameters $\in \mathsf{poly}(\lambda)$.

Assumptions.

For showing post-quantum security, we will rely on the pseudorandomness and correctness of the NTRU encryption scheme [HPS96, HPS98].

Definition B.4.7 (NTRU Encryption Assumption)**.** *Consider the NTRU encryption scheme parameterised by $n, q, p, \beta, \beta' \in \mathsf{poly}(\lambda)$ as given in Figure B.5.*

1. *We say that NTRU ciphertexts are w-pseudorandom if the following expression is negligible in λ for any PPT $\mathcal{A}$, arbitrary $m_i \in \mathcal{R}_p$ for $i \in \mathbb{Z}_w$, and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.*

$$\Pr\left[\mathcal{A}(\mathsf{pk}, \{\, c_i \,\}_{i \in \mathbb{Z}_w}) = 1 \middle| c_i \leftarrow \mathsf{Enc}(\mathsf{pk}, m_i)\right] - \Pr\left[\mathcal{A}(\mathsf{pk}, \{\, u_i \,\}_{i \in \mathbb{Z}_w}) = 1 \middle| u_i \leftarrow \mathcal{R}_q\right].$$

2. *Let $w, \alpha \in \mathsf{poly}(\lambda)$ be additional parameters. We say that NTRU decryption is (w, α)-correct if the following expression is negligible in λ for any PPT $\mathcal{A}$, $m_i \in \{0, 1\}$ for $i \in \mathbb{Z}_w$, and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.*

$$\Pr\left[\begin{array}{c} x_i \leftarrow \mathcal{A}(\mathsf{pk}, \{\, c_i \,\}_{i \in \mathbb{Z}_w}); \\ \forall i \in \mathbb{Z}_w,\ \|x_i\| \leq \alpha \wedge \mathsf{Dec}(\mathsf{sk}, \sum x_i \cdot c_i) \neq \sum_{i \in \mathbb{Z}_w} x_i \cdot m_i \end{array} \middle| c_i \leftarrow \mathsf{Enc}(\mathsf{pk}, m_i) \right].$$

The NTRU encryption assumption holds for the parameters n, q, w, α if there exist $p, \beta, \beta' \in \mathsf{poly}(\lambda)$ such that NTRU ciphertexts are w-pseudorandom and NTRU decryption is (w, α)-correct for these parameters.

The pseudorandomness of NTRU ciphertexts can be reduced to the decision NTRU assumption (asserting that NTRU public keys are pseudorandom) and the Ring-LWE assumption [CDH+20]. The decisional NTRU assumption can be dropped when $\beta \approx \sqrt{q}$ [SS11]. For any $\alpha \in \mathsf{poly}(\lambda)$, there exist parameters $n, q, p, \beta, \beta' \in \mathsf{poly}(\lambda)$ such that NTRU decryption is unconditionally correct [CDH+20].

Quantum Information.

A (pure) quantum state is a unit vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$. Hilbert spaces are commonly divided into registers, e.g., $\mathcal{H} = \mathcal{H}_0 \otimes \mathcal{H}_1$. A unitary operation is represented by a complex matrix U such that $UU^\dagger = \mathbf{I}$. The operation U transforms the pure state $|\psi\rangle$ to the pure state $U|\psi\rangle$. In this work, a quantum adversary is a family of quantum circuits $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ represented classically using some standard universal gate set. A quantum adversary is polynomial-size if there exists a polynomial p and some $\lambda_0 \in \mathbb{N}$ such that for all $\lambda > \lambda_0$ it holds that $|\mathcal{A}_\lambda| \leq p(\lambda)$.

Collapsing.

It is well known that the classical (computational) notion of binding is not meaningful against quantum attackers [Unr16, ARU14]. For compressing commitment schemes, where statistical binding is simply impossible, a more useful notion is that of *collapsing*. In the following, we adapt the definition of collapsing for hash functions [Unr16] to one for VCs. Essentially, our definition requires the commitment algorithm of the VC to be collapsing when viewed as a hash function. Note that our definition is weaker than that of [CMSZ22], who requires the collapsing property to hold with respect to positional openings.

Definition B.4.8 (Collapsing)**.** *A VC scheme Γ is said to be collapsing if for any QPT adversary $\mathcal{A}$ and any $w = \mathsf{poly}(\lambda)$ it holds that*

$$\left|\begin{array}{l} \Pr\left[\mathsf{CollapsExp}^0_{\Gamma,\mathcal{A}}(1^\lambda, 1^s, 1^w, 1^t) = 1\right] \\ -\ \Pr\left[\mathsf{CollapsExp}^1_{\Gamma,\mathcal{A}}(1^\lambda, 1^s, 1^w, 1^t) = 1\right] \end{array}\right| \leq \mathsf{negl}(\lambda).$$

where the experiment $\mathsf{CollapsExp}^b_{\Gamma,\mathcal{A}}(1^\lambda, 1^s, 1^w, 1^t)$ is defined as follows:

- *The challenger samples $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$ and sends it to $\mathcal{A}$.*
- *$\mathcal{A}$ replies with a classical message c (a commitment) and a quantum register $\mathcal{V}$, which contains strings $\mathbf{x} \in \mathcal{Z}^w$.*
- *Let U be the unitary that acts on $\mathcal{V}$ and some ancilla register and computes the bit $\left(c \stackrel{?}{=} \mathsf{Com}(\mathsf{pp}, \mathcal{V})\right)$, where the auxiliary output aux is suppressed. The challenger applies U to $\mathcal{V}$ and measures the ancilla register containing the output bit in the computational basis. If such bit is 0 abort the experiment, else apply $U^\dagger$.*
- *If $b = 0$ the challenger does nothing. If $b = 1$ the challenger measures the register $\mathcal{V}$ in the computational basis.*
- *Return the (possibly measured) register $\mathcal{V}$ to $\mathcal{A}$.*
- *$\mathcal{A}$ returns a bit which is also the output of the experiment.*

Somewhere Statistically Binding (SSB)

We introduce the notion of somewhere statistically binding (SSB) [HW15] for VCs. Similar to the treatment for collapsing above, our definition of SSB essentially requires that the commitment algorithm of the VC to be SSB as an ordinary commitment.

Definition B.4.9 (Somewhere Statistically Binding (SSB))**.** *A VC scheme Γ is said to be somewhere statistically binding (SSB) if there exists a binding setup algorithm $\mathsf{pp} \leftarrow \mathsf{BSetup}(1^\lambda, 1^s, 1^w, 1^t, i)$, which takes an additional input $i \in \mathbb{Z}_w$, such that the following properties are satisfied:*

- *(Mode Indistinguishability) For all $\lambda \in \mathbb{N}$, all $s, w, t = \mathsf{poly}(\lambda)$, and all $i \in \mathbb{Z}_w$ the following distributions are computationally indistinguishable*

$$\mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t) \approx \mathsf{BSetup}(1^\lambda, 1^w, 1^t, i).$$

- *(SSB) For all $\lambda \in \mathbb{N}$, $s, w, t = \mathsf{poly}(\lambda)$, $i \in \mathbb{Z}_w$, and $\mathsf{pp} \in \mathsf{BSetup}(1^\lambda, 1^s, 1^w, 1^t, i)$,*

$$\Pr\left[\exists \mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}^w : \begin{array}{r} (c_0, \mathsf{aux}_0) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x}_0) \\ \wedge\ (c_1, \mathsf{aux}_1) \leftarrow \mathsf{Com}(\mathsf{pp}, \mathbf{x}_1) \\ \wedge\ c_0 = c_1 \\ \wedge\ x_{0,i} \neq x_{1,i} \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Our central technique of achieving SSB is to replace entries of the public vector $\mathbf{v}$ with ciphertexts of (the provable variant of) the NTRU encryption scheme. Concretely, we construct $\mathsf{BSetup}(1^\lambda, 1^s, 1^w, 1^t, i)$ by setting v_i to be an NTRU ciphertext encrypting 1, while setting v_j to be an NTRU ciphertext encrypting 0 for all $j \neq i$. Since NTRU ciphertexts are indistinguishable from uniformly random $\mathcal{R}_q$ elements, mode indistinguishability follows. For the main SSB property, we notice that if two vectors $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}^w$ generate the same commitment, we have $\langle \mathbf{v}, \mathbf{x}_0 \rangle = \langle \mathbf{v}, \mathbf{x}_1 \rangle$. Since the NTRU encryption scheme is linearly homomorphic, the left-hand-side is a ciphertext encrypting $x_{0,i}$, while the right-hand-side is encrypting $x_{1,i}$. The correctness of NTRU then forces $x_{0,i} = x_{1,i}$.

Theorem B.4.10. *If the NTRU encryption assumption (Definition B.4.7) holds for n, q, w, α, the VC construction Γ in Figure B.4 is SSB.*

Proof. Following the treatment in Appendix B.4, we assume without loss of generality that $t = 1$ and omit the input 1^t to the setup algorithms. We begin by constructing the binding setup algorithm $\mathsf{BSetup}(1^\lambda, 1^s, 1^w, i)$ as follows, where $\mathbf{m}_i$ denotes the i-th unit vector.

Mode Indistinguishability. Fix any $i \in \mathbb{Z}_w$. To show that $\mathsf{Setup}(1^\lambda, 1^s, 1^w) \approx \mathsf{BSetup}(1^\lambda, 1^s, 1^w, i)$ it suffices to show that the distributions of $\mathbf{v}$ induced by the two algorithms are indistinguishable, which is immediately implied by the assumption that NTRU ciphertexts are w-pseudorandom.

$\mathsf{BSetup}(1^\lambda, 1^s, 1^w, i)$

$(\mathbf{A}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\eta, 1^\ell, q, \mathcal{R}, \beta)$
$\mathbf{t} \leftarrow \mathcal{T}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$v_i \leftarrow \mathsf{Enc}(\mathsf{pk}, 1)$
$v_j \leftarrow \mathsf{Enc}(\mathsf{pk}, 0),\ \forall j \in \mathbb{Z}_w \setminus \{ i \}$
$\mathbf{v} := (v_j : j \in \mathbb{Z}_w)$
$\mathbf{u}_g \leftarrow \mathsf{SampPre}(\mathsf{td}, g(\mathbf{v}) \cdot \mathbf{t}, \beta),\ \forall g \in \mathcal{G}$
return $\mathsf{pp} \coloneqq \left(\mathbf{A}, \mathbf{t}, \mathbf{v}, \{ \mathbf{u}_g \}_{g \in \mathcal{G}}\right)$

SSB. Fix any $i \in \mathbb{Z}_w$ and $\mathsf{pp} \in \mathsf{BSetup}(1^\lambda, 1^s, 1^w, i)$. We show that if $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}^w$ satisfy $\mathsf{Com}(\mathsf{pp}, \mathbf{x}_0) = \mathsf{Com}(\mathsf{pp}, \mathbf{x}_1)$ (suppressing aux), then it holds that $x_{0,i} = x_{1,i}$. Let sk be the NTRU secret key generated when generating the pp. Since $\mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}^w$, we have that $\|\mathbf{x}_0\| \leq \alpha$ and $\|\mathbf{x}_1\| \leq \alpha$. Let $c := \mathsf{Com}(\mathsf{pp}, \mathbf{x}_b) = \langle \mathbf{v}, \mathbf{x}_b \rangle \bmod q$. By the assumption that NTRU decryption is (w, α)-correct, it holds that $\mathsf{Dec}(\mathsf{sk}, c) = \langle \mathbf{m}_i, \mathbf{x}_i \rangle = x_{b,i}$ for $b \in \{0, 1\}$. Consequently, $x_{0,i} = x_{1,i}$. □

SSB Implies Collapsing.

We now show that an SSB VC is also collapsing. This implication was first shown in an oral presentation of Ma [Ma20] but, to the best of our knowledge, it does not formally appear in any prior work. For completeness, we present the proof below.

Theorem B.4.11. *An SSB VC Γ is collapsing.*

Proof. Let $\mathcal{V} = \mathcal{V}_0 \otimes \cdots \otimes \mathcal{V}_{w-1}$ denote the registers sent by the attacker in the collapsing experiment. The proof consists of a hybrid argument where we define the hybrids H_i for $i \in \{ 0, 1, \ldots, w \}$ to be the same experiment as $\mathsf{CollapsExp}^b_{\Gamma,\mathcal{A}}$ except that the challenger measures the registers $(\mathcal{V}_0, \ldots, \mathcal{V}_{i-1})$. Note that the hybrid H_0 corresponds to the original experiment with the bit $b = 0$, whereas hybrid H_w is identical to the original experiment with the bit set to $b = 1$. It therefore suffices to show that for all $i = [w]$ the hybrids H_{i-1} and H_i produce distributions that are computationally close. This is done by defining the following intermediate distributions:

- Hybrid G_0: This experiment is identical to H_{i-1}.
- Hybrid G_1: In this hybrid we compute the public parameters as $\mathsf{pp} \leftarrow \mathsf{BSetup}(1^\lambda, 1^s, 1^w, 1^t, i)$. By the mode indistinguishability of the setup algorithm, we can conclude that the view of the adversary is computationally indistinguishable from that induced by the previous hybrid.

- Hybrid G_2: This hybrid is identical to the previous one, except that the challenger additionally measures the i-th register $\mathcal{V}_i$. Let us analyse the content of the registers after the third step of the experiment. If the challenger aborts, then the adversary is not returned any register and therefore the views are trivially identical. On the other hand, if the challenger does not abort, then the state in the $\mathcal{V}$ register consists of

$$\chi = \sum_{\mathbf{x} \text{ s.t. } c=\mathsf{Com}(\mathsf{pp},\mathbf{x})} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$$

  where the amplitudes are suitably normalized and c is the classical string returned by $\mathcal{A}$. By the SSB property of the VC, it holds that, except with negligible probability, all pre-images of c have the same i-bit $\mathbf{x}_i$. Thus we can rewrite (up to a rearrangement of the registers)

$$\chi = \sum_{\mathbf{x} \text{ s.t. } c=\mathsf{Com}(\mathsf{pp},\mathbf{x})} \alpha_{\mathbf{x}} |\mathbf{x}\rangle = |\mathbf{x}_i\rangle \otimes \sum_{\mathbf{x} \text{ s.t. } c=\mathsf{Com}(\mathsf{pp},\mathbf{x})} \alpha_{\mathbf{x}} |\mathbf{x}_{-i}\rangle$$

  where $\mathbf{x}_{-i}$ denotes the vector $\mathbf{x}$ without the i-th bit $\mathbf{x}_i$. It follows that measuring the register $\mathcal{V}_i$ returns $\mathbf{x}_i$ with probability 1 and it does not disturb the state. Thus the adversary's view of this hybrid is statistically close to that of the previous one.

- Hybrid G_3: This is identical to the previous experiment, except that we undo the modification done in the G_1 (i.e., we sample the public parameters as $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^s, 1^w, 1^t)$). Computational indistinguishability follows by the same argument.

The proof is concluded by observing that the experiment G_3 is identical to H_i. □

APPENDIX C

Appendix to Chapter 4

C.1 Proofs for Foldable Structures

C.1.1 Proof of Lemma 4.6.1

Proof. By the definition of a foldable sequence, $n = \sum_{i=0}^{\ell} 2^i \cdot k_i$. Since $k_\ell \geq 1$, $k_i \leq k^*$ for all $i \in \{0, \ldots, \ell\}$, and $\sum_{i=0}^{\ell} 2^i = 2^{\ell+1} - 1$, we can derive $2^\ell \leq n < k^* \cdot 2^{\ell+1}$. The claim then follows. □

C.1.2 Power Sequence - Proof of Lemma 4.6.4

Proof. For the first claim, it suffices to show that the sequence of monomials $\mathbf{m} = (X, X^2, \ldots, X^n)$ in variable X is $(k_0, k_1, \ldots, k_\ell)$-foldable, and realise that $\mathbf{v}$ can be obtained by evaluating $\mathbf{m}$ at the point v. We construct a seed and a generator of $\mathbf{m}$ recursively as follows. Define a procedure, which on input a seed $\mathbf{m} = (X, X^2, \ldots, X^k)$ of length k and a generator $\mathbf{g} = \epsilon$, does the following:

- If $k \leq 2$, output $(\mathbf{m}, \mathbf{g})$.
- If $k > 2$ is odd (hence $k \geq 3$), write $k = 2 \cdot k' + 1$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{k'})$, $\ell' = 1$, $\mathbf{c}' = (X^{k'+1})$, and $r' = X^{k'+1}$.
- If k is even (hence $k \geq 4$), write $k = 2 \cdot k' + 2$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{k'})$, $\ell' = 1$, $\mathbf{c}' = (X^{k'+1}, X^{k'+2})$, and $r' = X^{k'+2}$.
- Let $\mathbf{g}' = (l', \mathbf{c}', r') \| \mathbf{g}$.
- Run the procedure on $(\mathbf{m}', \mathbf{g}')$.

It is easy to observe that running the above procedure on $(\mathbf{m}, \epsilon)$ finds a seed and a generator of $\mathbf{m}$ with the desired parameters.

For the generalised claim, we similarly define a procedure, which on input a seed $\mathbf{m} = (X, X^2, \ldots, X^{wk})$ of length wk and a generator $\mathbf{g} = \epsilon$, does the following:

- If $k \leq 2$, output $(\mathbf{m}, \mathbf{g})$.
- If $k > 2$ is odd (hence $k \geq 3$), write $k = 2 \cdot k' + 1$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{wk'})$, $\ell' = 1$, $\mathbf{c}' = (X^{wk'+1}, \ldots, X^{wk'+w})$, and $r' = X^{wk'+w}$.
- If k is even (hence $k \geq 4$), write $k = 2 \cdot k' + 2$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{wk'})$, $\ell' = 1$, $\mathbf{c}' = (X^{wk'+1}, \ldots, X^{wk'+2w})$, and $r' = X^{wk'+2w}$.
- Let $\mathbf{g}' = (l', \mathbf{c}', r') \| \mathbf{g}$.
- Run the procedure on $(\mathbf{m}', \mathbf{g}')$.

It is easy to observe that running the above procedure on $(\mathbf{m}, \epsilon)$ finds a seed and a generator of $\mathbf{m}$ with the desired parameters. □

C.1.3 Balanced Power Sequence - Proof of Lemma 4.6.5

Proof. It suffices to show that the sequence of monomials

$$\mathbf{m} = (X^{-n}, \ldots, X^{-2}, X^{-1}, X, X^2, \ldots, X^n)$$

in variable X is $(0, k_0, k_1, \ldots, k_\ell)$-foldable, and realise that $\mathbf{v}$ can be obtained by evaluating $\mathbf{m}$ at the point v.

Let $\hat{\mathbf{m}} = (X, X^2, \ldots, X^n)$, $\hat{\ell} = X^{-(n+1)}$ $\hat{\mathbf{c}} = \epsilon$ the empty vector, and $\hat{r} = 1$. Let $\hat{\mathbf{g}} = (\hat{l}, \hat{\mathbf{c}}, \hat{r})$. Clearly, $\hat{\mathbf{m}}$ is foldable with seed $\hat{\mathbf{m}}$ and generator $\hat{\mathbf{g}}$.

We next construct a generator of $\hat{\mathbf{m}}$ recursively as follows. Define a procedure, which on input a sequence $\mathbf{m}$ of length k and possibly partial generator $\mathbf{g}$, does the following:

- If $k = 1$, output $(\mathbf{m}, \mathbf{g})$.
- If $k > 1$ is odd (hence $k \geq 3$), write $k = 2 \cdot k' + 1$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{k'})$, $\ell' = 1$, $\mathbf{c}' = (X^{k'+1})$, and $r' = X^{k'+1}$.
- If k is even (hence $k \geq 2$), write $k = 2 \cdot k'$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{k'})$, $\ell' = 1$, $\mathbf{c}' = \epsilon$ the empty vector, and $r' = X^{k'}$.
- Let $\mathbf{g}' = (l', \mathbf{c}', r') \| \mathbf{g}$.
- Run the procedure on $(\mathbf{m}', \mathbf{g}')$.

It is easy to observe that running the above procedure on $(\hat{\mathbf{m}}, \hat{\mathbf{g}})$ finds a seed and a generator of $\mathbf{m}$ with the desired parameters. □

C.1.4 Compression Vector - Proof of Lemma 4.6.6

Proof. To show that $\mathbf{m}_\ell := \mathbf{m}$ is $(k_0, k_1, \ldots, k_\ell)$-foldable, it suffices to show that $\mathbf{m}_0, \ldots, \mathbf{m}_\ell$ induced by the given seed and generator as described in the procedure of Definition 4.6.1 each consists of distinct monomials.

By construction, $\mathbf{m}_\ell = (X_{\ell,1}, \ldots, X_{\ell,k_\ell})$ consists of distinct monomials. Suppose $\mathbf{m}_i$ consists of distinct monomials. Consider

$$\mathbf{m}_{i-1}^\intercal = (\mathbf{m}_i^\intercal, X_{i-1,1}, \ldots, X_{i-1,k_{i-1}}, X_{i-1,0} \cdot \mathbf{m}_i^\intercal).$$

By construction, none of the monomials in $\mathbf{m}_i^\intercal$ is a multiple of $X_{i-1,j}$ for any $j \in \{0, \ldots, k_{i-1}\}$. Therefore $\mathbf{m}_{i-1}$ consists of distinct monomials. The claim thus follows from induction.

Finally, the norm bound of $\mathbf{h}$ follows from the observation that each entry of $\mathbf{h}$ is a product of at most $\ell + 1$ entries of $\mathbf{x}$. □

C.2 Folding Argument for Type-1 Linear Relations

The protocol $\Pi_1^{\mathsf{fold}}.\langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}) \rangle$ consists of $\ell + 1$ rounds and makes use of the subtractive set $S \subset \mathcal{R}^\times$ mentioned in Section 4.3.1. Denote

$$(\mathbf{A}^{(0)}, \mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \alpha^{(0)}) := (\mathbf{A}, \mathbf{x}, \mathbf{y}, \alpha).$$

Let $n = \sum_{j=0}^{\ell} 2^j \cdot k_j$ be the binary representation of n, where $k_j \in \{0, 1\}$. For $i \in \{0, \ldots, \ell\}$, define $n_i := \sum_{j=i}^{\ell} 2^{j-i} \cdot k_j$. Then, for $i < \ell$, the i-th round of the protocol is as follows:

- Parse
  - $\mathbf{A}^{(i)}$ as $(\mathbf{A}_L^{(i)}, \mathbf{A}_c^{(i)}, \mathbf{A}_R^{(i)})$, and
  - $\mathbf{x}^{(i)}$ as $(\mathbf{x}_L^{(i)}, \mathbf{x}_c^{(i)}, \mathbf{x}_R^{(i)})$

  where $\mathsf{ncol}(\mathbf{A}_L^{(i)}) = \mathsf{ncol}(\mathbf{A}_R^{(i)}) = \mathsf{nrow}(\mathbf{x}_L^{(i)}) = \mathsf{nrow}(\mathbf{x}_R^{(i)}) = n_{i+1} \cdot w$. Note that $\mathsf{ncol}(\mathbf{A}_c^{(i)}) = k_i \cdot w$, meaning that $\mathbf{A}_c^{(i)}$ and $\mathbf{x}_c^{(i)}$ are empty when $k_i = 0$.
- $\mathcal{P}$ sends
  - $\mathbf{x}_c^{(i)}$ (if $k_i > 0$),
  - $\mathbf{y}_{LR}^{(i)} := \mathbf{A}_L^{(i)} \cdot \mathbf{x}_R^{(i)} \bmod q$, and
  - $\mathbf{y}_{RL}^{(i)} := \mathbf{A}_R^{(i)} \cdot \mathbf{x}_L^{(i)} \bmod q$.
- $\mathcal{V}$ samples $r_i \leftarrow S$ and sends r_i to $\mathcal{P}$.

- $\mathcal{P}$ computes the compressed witness

$$\mathbf{x}^{(i+1)} \coloneqq \mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i$$

- $\mathcal{P}$ and $\mathcal{V}$ compute the compressed statement

$$\begin{aligned}
\mathbf{A}^{(i+1)} &\coloneqq \mathbf{A}_L^{(i)} + \mathbf{A}_R^{(i)} \cdot r_i^{-1} \bmod q, \\
\mathbf{y}^{(i+1)} &\coloneqq \mathbf{y}^{(i)} - \mathbf{A}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{y}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{y}_{LR}^{(i)} \cdot r_i \bmod q \\
\alpha^{(i+1)} &\coloneqq 2 \cdot \alpha^{(i)} \cdot \gamma_{\mathcal{R}}.
\end{aligned}$$

In the ℓ-th (i.e. final) round, $\mathcal{P}$ sends $\mathbf{x}^{(\ell)}$ and $\mathcal{V}$ checks that

$$\mathbf{A}^{(\ell)} \cdot \mathbf{x}^{(\ell)} = \mathbf{y}^{(\ell)} \bmod q \qquad \text{and} \qquad \left\|\mathbf{x}^{(\ell)}\right\| \leq \alpha^{(\ell)}.$$

C.3 Proofs for Folding Arguments

C.3.1 Completeness - Proof of Theorem 4.7.1

Proof. The case where $n \leq 2$ is trivial. For $n > 2$, it is clear that for each i the checks $\left\|\mathbf{x}_c^{(i)}\right\| \leq \alpha^{(i)}$ and, if $k_i = 2$, $\begin{pmatrix}\mathbf{B} & \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} = \mathbf{y}_c^{(i)} \bmod q_0$, pass. It remains to show that, for each i, if

$$\begin{aligned}
\begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_i} \cdot \mathbf{x}^{(i)} &= \mathbf{y}^{(i)} \bmod q_0, \\
\mathbf{C}^{(i)} \cdot \mathbf{x}^{(i)} &= \mathbf{z}^{(i)} \bmod q_1,
\end{aligned} \qquad \text{and} \qquad \left\|\mathbf{x}^{(i)}\right\| \leq \alpha^{(i)},$$

then

$$\begin{aligned}
\begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}^{(i+1)} &= \mathbf{y}^{(i+1)} \bmod q_0, \\
\mathbf{C}^{(i+1)} \cdot \mathbf{x}^{(i+1)} &= \mathbf{z}^{(i+1)} \bmod q_1,
\end{aligned} \qquad \text{and} \qquad \left\|\mathbf{x}^{(i+1)}\right\| \leq \alpha^{(i+1)}.$$

First, by $\left(\begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_i}\right) \cdot \mathbf{x}^{(i)} = \mathbf{y}^{(i)} \bmod q_0$ we have

$$\begin{aligned}
\begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}_L^{(i)} + \begin{pmatrix}\mathbf{0} \\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} &= \mathbf{y}_L^{(i)} \bmod q_0, \\
\begin{pmatrix}\mathbf{B} \\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} + \begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}_R^{(i)} &= \mathbf{y}_R^{(i)} \bmod q_0.
\end{aligned}$$

It follows that

$$\begin{aligned}\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}^{(i+1)} &= \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot (\mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i)\\ &= \mathbf{y}_L^{(i)} - \begin{pmatrix}\mathbf{0}\\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} + \mathbf{y}_R^{(i)} \cdot r_i - \begin{pmatrix}\mathbf{B}\\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} \cdot r_i\\ &= \mathbf{y}^{(i+1)} \bmod q_0.\end{aligned}$$

Next, by $\mathbf{C}^{(i)} \cdot \mathbf{x}^{(i)} = \mathbf{z}^{(i)} \bmod q_1$ we have

$$\mathbf{C}_L^{(i)} \cdot \mathbf{x}_L^{(i)} + \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{C}_R^{(i)} \cdot \mathbf{x}_R^{(i)} = \mathbf{z}^{(i)} \bmod q_1.$$

Therefore

$$\begin{aligned}\mathbf{C}^{(i+1)} \cdot \mathbf{x}^{(i+1)} &= (\mathbf{C}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot r_i^{-1}) \cdot (\mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i)\\ &= \mathbf{C}_L^{(i)} \cdot \mathbf{x}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot \mathbf{x}_R^{(i)} + \mathbf{C}_R^{(i)} \cdot \mathbf{x}_L^{(i)} \cdot r_i^{-1} + \mathbf{C}_L^{(i)} \cdot \mathbf{x}_R^{(i)} \cdot r_i\\ &= \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{z}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{z}_{LR}^{(i)} \cdot r_i\\ &= \mathbf{z}^{(i+1)} \bmod q_1.\end{aligned}$$

Finally, since $\left\|\mathbf{x}^{(i)}\right\| \leq \alpha^{(i)}$, it follows that $\left\|\mathbf{x}^{(i+1)}\right\| = \left\|\mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i\right\| \leq 2 \cdot \alpha^{(i)} \cdot \gamma_{\mathcal{R}} = \alpha^{(i+1)}$. □

C.3.2 Special Soundness - Proof of Theorem 4.7.2

Proof. The case of $n \leq 2$ is trivial. Recall that $\alpha^{(i)} = (2\gamma_{\mathcal{R}})^i \cdot \alpha$. Let $\hat{\alpha}^{(\ell)} = \alpha^{(\ell)} = (2\gamma_{\mathcal{R}})^\ell \cdot \alpha$. For $i \in \{0, \ldots, \ell - 1\}$, define $\hat{\alpha}^{(i)} = 4\gamma_{\mathcal{R}}^3 \cdot \hat{\alpha}^{(i+1)}$, so that $\hat{\alpha}^{(0)} = (4\gamma_{\mathcal{R}}^3)^\ell \cdot \hat{\alpha}^{(\ell)} = (8\gamma_{\mathcal{R}}^4)^\ell \cdot \alpha$. In the following, assume that $n > 2$. We need to show that if $(\mathbf{x}_c^{(i)}, \mathbf{z}_{LR}^{(i)}, \mathbf{z}_{RL}^{(i)}, \mathbf{x}_0^{(i+1)}, \mathbf{x}_1^{(i+1)}, \mathbf{x}_2^{(i+1)})$ satisfies

$$\begin{aligned}&\begin{pmatrix}\mathbf{B} & \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} = \mathbf{y}_c^{(i)} \bmod q_0 \text{ if } k_i = 2, && \left\|\mathbf{x}_c^{(i)}\right\| \leq \alpha^{(i)},\\ &\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}_j^{(i+1)} = \mathbf{y}_j^{(i+1)} \bmod q_0, && \\ &\mathbf{C}_j^{(i+1)} \cdot \mathbf{x}_j^{(i+1)} = \mathbf{z}_j^{(i+1)} \bmod q_1, && \text{and} \quad \left\|\mathbf{x}_j^{(i+1)}\right\| \leq \hat{\alpha}^{(i+1)},\end{aligned}$$

where

$$\begin{aligned}\mathbf{C}_j^{(i+1)} &= \mathbf{C}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot r_{i,j}^{-1} \bmod q_1,\\ \mathbf{y}_j^{(i+1)} &= \mathbf{y}_L^{(i)} + \mathbf{y}_R^{(i)} \cdot r_{i,j} - \begin{pmatrix}\mathbf{B} \cdot r_{i,j}\\ \mathbf{0}\\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} \bmod q_0\\ \mathbf{z}_j^{(i+1)} &= \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{z}_{RL}^{(i)} \cdot r_{i,j}^{-1} + \mathbf{z}_{LR}^{(i)} \cdot r_{i,j} \bmod q_1\end{aligned}$$

for distinct challenges $r_{i,0}, r_{i,1}, r_{i,2} \in S$, then we can extract $\mathbf{x}^{(i)}$ satisfying

$$\begin{aligned} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_i} \cdot \mathbf{x}^{(i)} &= \mathbf{y}^{(i)} \bmod q_0, \\ \mathbf{C}^{(i)} \cdot \mathbf{x}^{(i)} &= \mathbf{z}^{(i)} \bmod q_1, \end{aligned} \qquad \text{and} \qquad \left\|\mathbf{x}^{(i)}\right\| \leq \hat{\alpha}^{(i)}.$$

Let

$$\mathbf{X} \coloneqq \begin{pmatrix} \mathbf{x}_0^{(i+1)} & \mathbf{x}_1^{(i+1)} & \mathbf{x}_2^{(i+1)} \\ \mathbf{x}_0^{(i+1)} \cdot r_{i,0}^{-1} & \mathbf{x}_1^{(i+1)} \cdot r_{i,1}^{-1} & \mathbf{x}_2^{(i+1)} \cdot r_{i,2}^{-1} \end{pmatrix} \quad \text{and} \quad \mathbf{V} \coloneqq \begin{pmatrix} r_{i,0}^{-1} & r_{i,1}^{-1} & r_{i,2}^{-1} \\ 1 & 1 & 1 \\ r_{i,0} & r_{i,1} & r_{i,2} \end{pmatrix}.$$

From the hypothesis, we can derive the following relations:

$$\left(\begin{array}{c|c} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} & \\ \hline & \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \end{array}\right) \cdot \mathbf{X} = \begin{pmatrix} \mathbf{0} & \mathbf{y}_L^{(i)} - \begin{pmatrix}\mathbf{0}\\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} & \mathbf{y}_R^{(i)} - \begin{pmatrix}\mathbf{B}\\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} \\ \mathbf{y}_L^{(i)} - \begin{pmatrix}\mathbf{0}\\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} & \mathbf{y}_R^{(i)} - \begin{pmatrix}\mathbf{B}\\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} & \mathbf{0} \end{pmatrix} \cdot \mathbf{V} \bmod$$

$$\left(\begin{array}{c|c} \mathbf{C}_L^{(i)} & \mathbf{C}_R^{(i)} \end{array}\right) \cdot \mathbf{X} = \left(\mathbf{z}_{RL}^{(i)} \quad \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} \quad \mathbf{z}_{LR}^{(i)}\right) \cdot \mathbf{V} \bmod q_1.$$

Since $\det(\mathbf{V}) = r_{i,0}^{-1} \cdot r_{i,1}^{-1} \cdot r_{i,2}^{-1} \cdot (r_{i,0} - r_{i,1}) \cdot (r_{i,1} - r_{i,2}) \cdot (r_{i,2} - r_{i,0})$ and S is subtractive, $\mathbf{V}$ is invertible. Let

$$\begin{pmatrix} \mathbf{x}_L^{(i)} \\ \mathbf{x}_R^{(i)} \end{pmatrix} \coloneqq \mathbf{X} \cdot \mathbf{V}^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

We have

$$\left(\begin{array}{c|c} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} & \\ \hline & \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \end{array}\right) \cdot \begin{pmatrix} \mathbf{x}_L^{(i)} \\ \mathbf{x}_R^{(i)} \end{pmatrix} = \begin{pmatrix} \mathbf{y}_L^{(i)} - \begin{pmatrix}\mathbf{0}\\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} \\ \mathbf{y}_R^{(i)} - \begin{pmatrix}\mathbf{B}\\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} \end{pmatrix} \bmod q_0,$$

$$\left(\begin{array}{c|c} \mathbf{C}_L^{(i)} & \mathbf{C}_R^{(i)} \end{array}\right) \cdot \begin{pmatrix} \mathbf{x}_L^{(i)} \\ \mathbf{x}_R^{(i)} \end{pmatrix} = \left(\mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)}\right) \bmod q_1,$$

or equivalently

$$\begin{aligned} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_i} \cdot \mathbf{x}^{(i)} &= \mathbf{y}^{(i)} \bmod q_0 \\ \mathbf{C}^{(i)} \cdot \mathbf{x}^{(i)} &= \mathbf{z}^{(i)} \bmod q_1 \end{aligned}$$

where $\mathbf{x}^{(i)} = (\mathbf{x}_L^{(i)}, \mathbf{x}_c^{(i)}, \mathbf{x}_R^{(i)})$.

It remains to show that $\left\|\mathbf{x}^{(i)}\right\| \leq \hat{\alpha}^{(i)}$. Note that

$$\underbrace{\mathbf{V}^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}}_{\mathbf{w}_i} = \begin{pmatrix} \frac{r_{i,0} \cdot (r_{i,1} + r_{i,2})}{(r_{i,0} - r_{i,1}) \cdot (r_{i,2} - r_{i,0})} \\ \frac{r_{i,1} \cdot (r_{i,2} + r_{i,0})}{(r_{i,0} - r_{i,1}) \cdot (r_{i,1} - r_{i,2})} \\ \frac{r_{i,2} \cdot (r_{i,0} + r_{i,1})}{(r_{i,1} - r_{i,2}) \cdot (r_{i,2} - r_{i,0})} \end{pmatrix}$$

where each entry can be simplified to be of the form

$$\frac{-(\zeta^a - 1) \cdot (\zeta^b + \zeta^c - 2)}{(\zeta^a - \zeta^b) \cdot (\zeta^a - \zeta^c)}.$$

By a routine calculation (see e.g. [AL21, Proposition 11]), the norm of the above and hence $\|\mathbf{w}_i\|$ can be upper bounded by $4\gamma_{\mathcal{R}}$. Therefore $\left\|\mathbf{x}^{(i)}\right\| \leq 4\gamma_{\mathcal{R}}^3 \cdot \hat{\alpha}^{(i+1)} = \hat{\alpha}^{(i)}$. □

C.3.3 Efficiency - Proof of Theorem 4.7.3

Proof. Note that $\log |\mathcal{R}_{q_0}| < \log |\mathcal{R}_{q_1}| = \log q_1^{\varphi(\rho)} = O_\lambda(\log q_1) = O_\lambda(\log n)$, and an $\mathcal{R}_{q_1}$ operation takes at most $O_\lambda(\log^2 n)$ bit operations. It is easy to verify that the prover computes $O_\lambda(n)$ operations over $\mathcal{R}_{q_0}$ or $\mathcal{R}_{q_1}$, which takes $O_\lambda(n \cdot \log^2 n)$ time, and that $O_\lambda(\ell)$ elements of $\mathcal{R}_{q_0}$ or $\mathcal{R}_{q_1}$ are being communicated, for which the overall description size is at most $O_\lambda(\log^2 n)$. To analyse the computation cost of the verifier, we break down the computation steps, consisting of $O_\lambda(\ell + \sum_{i=0}^{\ell} k_i) = O_\lambda(\log n)$ operations over $\mathcal{R}_{q_0}$ or $\mathcal{R}_{q_1}$, which take time $O_\lambda(\log^3 n)$, into three parts.

First, $O_\lambda(\sum_{i=0}^{\ell-i} k_i)$ operations over $\mathcal{R}_{q_1}$ are contributed by the computation of

$$\mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} \bmod q_1, \ i \in \{0, \ldots, \ell - 1\}.$$

Second, $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_1}$ are contributed by the recursive computation of

$$\mathbf{C}^{(i+1)} = \mathbf{C}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot r_i^{-1} \bmod q_1, \ i \in \{0, \ldots, \ell - 1\}.$$

Since $\mathbf{C}$ is $(k_0, \ldots, k_\ell)$-block-foldable with block-size w, there exists $\mathsf{poly}(\lambda)$-size matrices $\mathbf{M}_\ell$ over $\mathcal{R}_{q_1}^{k_\ell}$ and $(\mathbf{L}_i, \mathbf{R}_i)_{i=0}^{\ell-1}$ over $\mathcal{R}_{q_1}$ such that

$$\mathbf{C}^{(\ell)} = (\mathbf{L}_0 + \mathbf{R}_0 \cdot r_0^{-1}) \circ \ldots \circ (\mathbf{L}_{\ell-1} + \mathbf{R}_{\ell-1} \cdot r_{\ell-1}^{-1}) \circ \mathbf{M}_\ell \bmod q_1.$$

Computing $\mathbf{C}^{(\ell)}$ this way requires $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_1}$.

Third, another $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_0}$ are contributed by the recursive computation of

$$\mathbf{y}^{(i+1)} = \mathbf{y}_L^{(i)} + \mathbf{y}_R^{(i)} \cdot r_i - \begin{pmatrix} \mathbf{B} \cdot r_i \\ \mathbf{0} \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_c^{(i)} \bmod q_0$$

for $i \in \{0, \ldots, \ell - 1\}$. Since $\mathbf{y}$ is $(k_0 - 1, \ldots, k_{\ell-1}, k_\ell + 1)$-block-foldable with block-size h_0, there exists $\mathsf{poly}(\lambda)$-size vectors $\mathbf{m}_\ell$ over $\mathcal{R}_{q_0}^{k_\ell+1}$ and $(\mathbf{l}_i, \mathbf{r}_i)_{i=0}^{\ell-1}$ over $\mathcal{R}_{q_0}$ such that

$$\mathbf{y}^{(\ell)} = (\mathbf{l}_0 + \mathbf{r}_0 \cdot r_0^{-1}) \circ \ldots \circ (\mathbf{l}_{\ell-1} + \mathbf{r}_{\ell-1} \cdot r_{\ell-1}^{-1}) \circ \mathbf{m}_\ell - \sum_{i=0}^{\ell-1} (\mathbf{A} + \mathbf{B} \cdot r_i) \cdot \mathbf{x}_c^{(i)} \bmod q_0.$$

Computing $\mathbf{y}^{(\ell)}$ this way requires $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_0}$.

Last, the remaining $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_1}$ are contributed by the recursive computation of

$$\begin{aligned} \mathbf{z}^{(i+1)} &= \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{z}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{z}_{LR}^{(i)} \cdot r_i \bmod q_1, \text{ and} \\ \alpha^{(i+1)} &= 2 \cdot \alpha^{(i)} \cdot \gamma_{\mathcal{R}} \end{aligned}$$

for $i \in \{0, \ldots, \ell - 1\}$ and well as the final check

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow k_\ell} \cdot \mathbf{x}^{(\ell)} = \mathbf{y}^{(\ell)} \bmod q_0, \qquad \text{and} \qquad \left\| \mathbf{x}^{(\ell)} \right\| \leq \alpha^{(\ell)}.$$
$$\mathbf{C}^{(\ell)} \cdot \mathbf{x}^{(\ell)} = \mathbf{z}^{(\ell)} \bmod q_1,$$

□

C.4 Knowledge-based Argument for Well-formedness of vSIS Commitments

Let $\mathcal{R}, s, \eta, m, q_1, q_3, \alpha, \beta, \delta, \mathcal{T}$ depend on λ. Using the lattice trapdoor algorithms (Section 4.3.2) parametrised by (η, m, q_3, β), in Figure C.1 we give a formal description of $\Pi_1^{\mathtt{know}}$.

C.5 Proofs for Knowledge-based Arguments

C.5.1 Completeness - Proof of Theorem 4.8.1

Proof. Condition b_0. We first consider the condition b_0 in the verification algorithm. Recall that $c_0 = c_{0,0} + c_{0,1} \bmod q_3$ where

$$\begin{aligned} c_{0,0} &= \bar{c}_{\mathbf{M}} \cdot c_{\mathbf{x}} + \bar{c}_{q_0} \cdot c_{\mathbf{r}} - \hat{c}_{\mathbf{y}} \bmod q_3, \\ c_{0,1} &= \bar{c}_{\mathbf{I}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{x}} \cdot c_{\mathbf{I}} \bmod q_3. \end{aligned}$$

$\underline{\mathsf{Setup}(1^\lambda, \mathsf{pp})}$

$v \leftarrow \mathsf{pp};\ \mathbf{t} \leftarrow \mathcal{T}$

$(\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$

$\mathbf{u}_i \leftarrow \mathsf{SampPre}(\mathsf{td}, \mathbf{t} \cdot v^i),\ \forall i \in \pm[s]$

$\mathsf{pp} := v$

$\mathsf{crs} := \left(\mathbf{D}, \mathbf{t}, v, (\mathbf{u}_i)_{i \in \pm[s]}\right)$

return crs

$\underline{\mathsf{Prove}(\mathsf{crs}, (\epsilon, c_{\mathbf{z}}), \mathbf{z})}$

$\mathbf{u} := \sum_{i \in \pm[s]} \mathbf{u}_i \cdot z_i$

return $\pi := \mathbf{u}$

$\underline{\mathsf{PreVerify}(\mathsf{crs}, \epsilon)}$

return $\mathsf{pp}_\epsilon := (\mathbf{D}, \mathbf{t})$

$\underline{\mathsf{Verify}(\mathsf{crs}_\epsilon, c_{\mathbf{z}}, \pi)}$

$b_0 := \left(\mathbf{D} \cdot \mathbf{u} \stackrel{?}{=} \mathbf{t} \cdot c_{\mathbf{z}} \bmod q_3\right)$

$b_1 := \left(\|\mathbf{u}\| \stackrel{?}{\leq} \delta\right)$

return $b_0 \wedge b_1$

Figure C.1: Our argument system $\Pi_1^{\mathtt{know}}$.

Substituting the expressions of each component, we have

$$\begin{aligned} c_{0,0} &= \mathbf{f}_0^\mathsf{T} \cdot \mathbf{M} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x} + \mathbf{f}_0^\mathsf{T} \cdot q_0 \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\mathsf{T} \cdot \mathbf{r} - \mathbf{f}_0^\mathsf{T} \cdot \mathbf{y} \bmod q_3 \\ &= \mathbf{f}_0^\mathsf{T} \cdot (\mathbf{M} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x} + q_0 \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\mathsf{T} \cdot \mathbf{r} - \mathbf{y}) \bmod q_3 \\ &= \mathbf{f}_0^\mathsf{T} \cdot (\mathbf{M} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} - \mathbf{I}_s) \cdot \mathbf{x} + q_0 \cdot (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\mathsf{T} - \mathbf{I}_t) \cdot \mathbf{r}) \bmod q_3 \\ &= \sum_{i,j \in [s], k \in [t], i \neq j} f_{0,k} \cdot M_{k,i} \cdot v^{j-i} \cdot x_j + \sum_{i,j,k \in [t], i \neq j} f_{0,k} \cdot q_0 \cdot v^{j-i} \cdot r_j \bmod q_3 \end{aligned}$$

where the last equality is due to $\mathbf{M} \cdot \mathbf{x} + q_0 \cdot \mathbf{r} = \mathbf{y}$, and

$$\begin{aligned} c_{0,1} &= \mathbf{f}_1^\mathsf{T} \cdot \mathbf{I} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x} - \mathbf{x}^\mathsf{T} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{I} \cdot \mathbf{f}_1 \bmod q_3 \\ &= \mathbf{f}_1^\mathsf{T} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x} - \mathbf{f}_1^\mathsf{T} \cdot \mathbf{v} \cdot (\bar{\mathbf{v}} \circ \mathbf{h})^\mathsf{T} \cdot \mathbf{x} \bmod q_3 \\ &= \mathbf{f}_1^\mathsf{T} \cdot ((\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^\mathsf{T} - \mathbf{v} \cdot (\bar{\mathbf{v}} \circ \mathbf{h})^\mathsf{T}) \cdot \mathbf{x} \bmod q_3 \\ &= \sum_{i,j \in [s]} f_{1,i} \cdot (h_i \cdot v^{j-i} - v^{i-j} \cdot h_j) \cdot x_j \bmod q_3 \\ &= \sum_{i,j \in [s]} f_{1,j} \cdot h_j \cdot v^{i-j} \cdot x_i - \sum_{i,j \in [s]} f_{1,i} \cdot v^{i-j} \cdot h_j \cdot x_j \bmod q_3 \\ &= \sum_{i,j \in [s]} v^{i-j} \cdot h_j \cdot (f_{1,j} \cdot x_i - f_{1,i} \cdot x_j) \bmod q_3. \end{aligned}$$

Since

$$\mathbf{D}_0 \cdot \mathbf{u}_{0,i} = \mathbf{t}_0 \cdot v^i \bmod q_3$$

for all $i \in \pm[\max\{s,t\}]$, it follows that

$$\mathbf{D}_0 \cdot \mathbf{u}_0 = \mathbf{t}_0 \cdot c_0 \bmod q_3.$$

Furthermore, we observe the following norm bounds:

1. $\|\mathbf{u}_{0,j}\| \leq \beta$ for all $j \in \pm[\max\{s,t\}]$,
2. $\|\mathbf{h}\| \leq q_1/2$,
3. $\|\mathbf{f}_0\|, \|\mathbf{f}_1\| \leq q_2/2$,
4. $\|\mathbf{M}\|, \|\mathbf{y}\| \leq q_0/2$,
5. $\|x\| \leq \alpha$, and
6. $\|r\| \leq \frac{1}{q_0} \cdot (s \cdot \|M\| \cdot \|\mathbf{x}\| \cdot \gamma_{\mathcal{R}} + \|\mathbf{y}\|) \leq s \cdot \alpha \cdot \gamma_{\mathcal{R}}$.

Therefore

$$\begin{aligned}\|\mathbf{u}_0\| &\leq s^2 \cdot t \cdot q_2 \cdot q_0 \cdot \beta \cdot \alpha \cdot \gamma_{\mathcal{R}}^3 + t^3 \cdot q_2 \cdot q_0 \cdot \beta \cdot s \cdot \alpha \cdot \gamma_{\mathcal{R}}^3 + s^2 \cdot \beta \cdot q_1 \cdot q_2 \cdot \alpha \cdot \gamma_{\mathcal{R}}^3 \\ &\leq (s+t)^4 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3 \\ &\leq \delta_0.\end{aligned}$$

Conditions b_1, b_2, and b_3. We next consider the conditions b_1, b_2, and b_3 in the verification algorithm. Clearly, it holds that

$$\begin{aligned}\mathbf{D}_1 \cdot \mathbf{u}_1 &= \mathbf{D}_1 \cdot \left(\sum_{j \in [s]} \mathbf{u}_{1,j} \cdot x_j \right) = \sum_{j \in [s]} \mathbf{t}_1 \cdot v^j \cdot x_j = \mathbf{t}_1 \cdot c_{\mathbf{x}} \bmod q_3, \\ \mathbf{D}_2 \cdot \mathbf{u}_2 &= \mathbf{D}_2 \cdot \left(\sum_{j \in [s]} \mathbf{u}_{2,-j} \cdot h_j \cdot x_j \right) = \sum_{j \in [s]} \mathbf{t}_2 \cdot v^{-j} \cdot h_j \cdot x_j = \mathbf{t}_2 \cdot \bar{c}_{\mathbf{x}} \bmod q_3, \\ \mathbf{D}_3 \cdot \mathbf{u}_3 &= \mathbf{D}_3 \cdot \left(\sum_{j \in [t]} \mathbf{u}_{3,j} \cdot r_j \right) = \sum_{j \in [t]} \mathbf{t}_3 \cdot v^j \cdot r_j = \mathbf{t}_3 \cdot c_{\mathbf{r}} \bmod q_3.\end{aligned}$$

Furthermore, since $\|\mathbf{u}_{1,j}\| \leq \beta$ for $j \in [s]$, $\|\mathbf{u}_{2,j}\| \leq \beta$ for $j \in -[s]$, $\|\mathbf{u}_{3,j}\| \leq \beta$ for $j \in [t]$, $\|\mathbf{h}\| \leq q_1/2$, $\|\mathbf{x}\| \leq \alpha$, and $\|r\| \leq s \cdot \alpha \cdot \gamma_{\mathcal{R}}$, we have

$$\begin{aligned}\|\mathbf{u}_1\| &\leq s \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}} \leq \delta_1, \\ \|\mathbf{u}_2\| &\leq s \cdot q_1 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}} \leq \delta_2, \\ \|\mathbf{u}_3\| &\leq s^2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2 \leq \delta_3.\end{aligned}$$

Putting everything together yields the claim. □

C.5.2 Knowledge Soundness - Proof of Theorem 4.8.2

Proof. Fix a PPT prover $\mathcal{P}^*$. Consider an algorithm $\mathcal{B}_1 = \mathcal{B}^{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, runs $\pi \leftarrow \mathcal{P}^*(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, parses $c_{\mathbf{x}}$ from stmt and $\mathbf{u}_1$ from π, and outputs $(c_{\mathbf{x}}, \mathbf{u}_1)$. Similarly, consider the algorithms $\mathcal{B}_2 = \mathcal{B}^{\mathcal{P}^*}$ and $\mathcal{B}_3 = \mathcal{B}^{\mathcal{P}^*}$ which do almost the same, except that $\mathcal{B}_2 = \mathcal{B}^{\mathcal{P}^*}$ parses $\bar{c}_{\mathbf{x}}$ from stmt and $\mathbf{u}_2$ from π and outputs $(\bar{c}_{\mathbf{x}}, \mathbf{u}_2)$, and $\mathcal{B}_3 = \mathcal{B}^{\mathcal{P}^*}$ parses $c_{\mathbf{r}}$ from stmt and $\mathbf{u}_3$ from π and outputs $(c_{\mathbf{r}}, \mathbf{u}_3)$. Let $\mathcal{E}^{k\text{-}R\text{-}\mathsf{ISIS},1}_{\mathcal{B}_1}$, $\mathcal{E}^{k\text{-}R\text{-}\mathsf{ISIS},2}_{\mathcal{B}_2}$, and $\mathcal{E}^{k\text{-}R\text{-}\mathsf{ISIS},3}_{\mathcal{B}_3}$ be the knowledge extractors whose existence are guaranteed by Assumptions 1, 2, and 3. Define an extractor $\mathcal{E}_{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, does the following:

- run $\mathbf{x}_1^\dagger \leftarrow \mathcal{E}^{k\text{-}R\text{-}\mathsf{ISIS},1}_{\mathcal{B},1}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{x}_2^\dagger \leftarrow \mathcal{E}^{k\text{-}R\text{-}\mathsf{ISIS},2}_{\mathcal{B},2}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{r}^\dagger \leftarrow \mathcal{E}^{k\text{-}R\text{-}\mathsf{ISIS},3}_{\mathcal{B},3}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- checks that $\mathbf{x}_1^\dagger \circ \mathbf{h} = \mathbf{x}_2^\dagger$,
- checks that $\mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger = \mathbf{y}$, and
- outputs $\mathbf{x}^\dagger := \mathbf{x}_1^\dagger$ if both checks pass.

Fix any adversary $\mathcal{A}$ and consider the following experiment Exp:

$\mathsf{Exp}(1^\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Gen}^{\mathtt{unstr}}(1^\lambda)$
$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp})$
$(\mathsf{stmt}, \mathsf{wit}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{crs})$
$(\pi, \mathsf{wit}^\dagger) \leftarrow (\mathcal{P}^* | \mathcal{E}_{\mathcal{P}^*})(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$
$\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$
return $\mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}, \pi) = 1 \wedge (\mathsf{stmt}, \mathsf{wit}^\dagger) \notin \Psi_{\mathsf{pp}}$

We claim that $\Pr[\mathsf{Exp}(1^\lambda) = 1] \leq \mathsf{negl}(\mathsf{negl})$, which proves the theorem.

To prove the claim, consider a modified experiment Exp' where in the setup $\mathsf{Setup}(1^\lambda, \mathsf{pp})$ the matrices $\mathbf{D}_0, \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ are sampled uniformly at random and the $\mathsf{SampPre}$ steps are replaced with sampling from SampD subject to the appropriate constraints. By the properties of $(\mathsf{TrapGen}, \mathsf{SampD}, \mathsf{SampPre})$, Exp' is statistically close to Exp. Therefore it suffices to show that $\Pr[\mathsf{Exp}'(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

We now examine $\mathsf{wit}^\dagger$ generated during the execution of $\mathsf{Exp}'(1^\lambda)$. Parse $\mathsf{stmt} = (\mathbf{M}, \mathbf{y}, c_{\mathbf{x}}, \bar{c}_{\mathbf{x}})$ and $\mathsf{wit}^\dagger = \mathbf{x}^\dagger$. First, suppose that $\mathcal{E}_{\mathcal{P}^*}$ returns something, i.e. $\mathbf{x}_2^\dagger = \mathbf{x}_1^\dagger \circ \mathbf{h}$

and $\mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger = \mathbf{y}$, then by Conditions b_1, b_2, and b_3 of the verification algorithm and Assumptions 1, 2, and 3 we have

$$\begin{aligned} c_{\mathbf{x}} &= \mathbf{v}^\intercal \cdot \mathbf{x}_1^\dagger \bmod q_3, & \left\|\mathbf{x}_1^\dagger\right\| &\leq \alpha_1^* \\ \bar{c}_{\mathbf{x}} &= \bar{\mathbf{v}}^\intercal \cdot \mathbf{x}_2^\dagger \bmod q_3, & \left\|\mathbf{x}_2^\dagger\right\| &\leq \alpha_2^* \\ c_{\mathbf{r}} &= \mathbf{v}_t^\intercal \cdot \mathbf{r}^\dagger \bmod q_3, \quad \text{and} & \left\|\mathbf{r}^\dagger\right\| &\leq \alpha_3^*. \end{aligned}$$

It remains to show that $\mathbf{x}_1^\dagger \circ \mathbf{h} = \mathbf{x}_2^\dagger$ and $\mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger = \mathbf{y}$, so that $\mathcal{E}_{\mathcal{P}^*}$ returns something, with overwhelming probability.

Examining the condition b_0 in the verification algorithm, we observe

$$\begin{aligned} &\mathbf{D}_0 \cdot \mathbf{u}_0 \\ &= \mathbf{t}_0 \cdot (\bar{c}_{\mathbf{M}} \cdot c_{\mathbf{x}} + \bar{c}_{q_0} \cdot c_{\mathbf{r}} - \hat{c}_{\mathbf{y}} + \bar{c}_{\mathbf{I}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{x}} \cdot c_{\mathbf{I}}) \bmod q_3 \\ &= \mathbf{t}_0 \cdot (\mathbf{f}_0^\intercal \cdot \mathbf{M} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\intercal \cdot \mathbf{x}_1^\dagger + \mathbf{f}_0^\intercal \cdot q_0 \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\intercal \cdot \mathbf{r}^\dagger \\ &\qquad + \mathbf{f}_1^\intercal \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^\intercal \cdot \mathbf{x}_1^\dagger - \mathbf{f}_0^\intercal \cdot \mathbf{y} - \bar{\mathbf{v}}^\intercal \cdot \mathbf{x}_2^\dagger \cdot \mathbf{v}^\intercal \cdot \mathbf{f}_1) \bmod q_3 \\ &= \mathbf{t}_0 \cdot \mathbf{f}_0^\intercal \cdot (\mathbf{M} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\intercal \cdot \mathbf{x}_1^\dagger + q_0 \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\intercal \cdot \mathbf{r}^\dagger - \mathbf{y}) \\ &\qquad + \mathbf{t}_0 \cdot \mathbf{f}_1^\intercal \cdot (\mathsf{diag}(\mathbf{h}) \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\intercal \cdot \mathbf{x}_1^\dagger - \mathbf{v} \cdot \bar{\mathbf{v}}^\intercal \cdot \mathbf{x}_2^\dagger) \bmod q_3 \\ &= \mathbf{t}_0 \cdot \mathbf{f}_0^\intercal \cdot (\mathbf{M} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^\intercal - \mathbf{I}_s) \cdot \mathbf{x}_1^\dagger + q_0 \cdot (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\intercal - \mathbf{I}_t) \cdot \mathbf{r}^\dagger + (\mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger - \mathbf{y})) \\ &\qquad + \mathbf{t}_0 \cdot \mathbf{f}_1^\intercal \cdot (\mathsf{diag}(\mathbf{h}) \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^\intercal - \mathbf{I}_s) \cdot \mathbf{x}_1^\dagger - (\mathbf{v} \cdot \bar{\mathbf{v}}^\intercal - \mathbf{I}_s) \cdot \mathbf{x}_2^\dagger + (\mathbf{h} \circ \mathbf{x}_1^\dagger - \mathbf{x}_2^\dagger)) \bmod q_3. \end{aligned}$$

Let

$$\begin{aligned} \mathbf{u}_0^\dagger \coloneqq &\sum_{i,j \in [s], k \in [t] : i \neq j} f_{0,k} \cdot M_{k,i} \cdot \mathbf{u}_{0,j-i} \cdot x_{1,j}^\dagger + \sum_{i,j,k \in [t] : i \neq j} f_{0,k} \cdot q_0 \cdot \mathbf{u}_{0,j-i} \cdot r_j^\dagger \\ &+ \sum_{i,j \in [s] : i \neq j} f_{1,i} \cdot h_i \cdot \mathbf{u}_{0,j-i} \cdot x_{1,j}^\dagger + \sum_{i,j \in [s] : i \neq j} f_{1,i} \cdot \mathbf{u}_{0,i-j} \cdot x_{2,j}^\dagger, \\ \mathbf{e}_0^\dagger \coloneqq &\ \mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger - \mathbf{y}, \\ \mathbf{e}_1^\dagger \coloneqq &\ \mathbf{h} \circ \mathbf{x}_1^\dagger - \mathbf{x}_2^\dagger. \end{aligned}$$

We have

$$\mathbf{D}_0 \cdot (\mathbf{u}_0 - \mathbf{u}_0^\dagger) = \mathbf{t}_0 \cdot (\mathbf{f}_0^\intercal \cdot \mathbf{e}_0^\dagger + \mathbf{f}_1^\intercal \cdot \mathbf{e}_1^\dagger) \bmod q_3.$$

Suppose, contrary to our claim, that $(\mathbf{e}_0^\dagger, \mathbf{e}_1^\dagger) \neq \mathbf{0}$ with non-negligible probability. Then one (or both) of the following must be true:

1. $\mathbf{f}_0^\intercal \cdot \mathbf{e}_0^\dagger + \mathbf{f}_1^\intercal \cdot \mathbf{e}_1^\dagger = \mathbf{0}$ with non-negligible probability.
2. $\mathbf{f}_0^\intercal \cdot \mathbf{e}_0^\dagger + \mathbf{f}_1^\intercal \cdot \mathbf{e}_1^\dagger \neq \mathbf{0}$ with non-negligible probability.

If Case (i) is true, then we also have with non-negligible probability

$$\mathbf{f}_0^{\mathsf{T}} \cdot \mathbf{e}_0^{\dagger} + \mathbf{f}_1^{\mathsf{T}} \cdot \mathbf{e}_1^{\dagger} = \mathbf{0} \bmod q_2.$$

Note that

$$\begin{aligned}\left\|\mathbf{e}_0^{\dagger}\right\| &\leq s \cdot q_0/2 \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}} + q_0 \cdot \alpha_3^* + q_0/2 \leq s \cdot q_0 \cdot \alpha^* \cdot \gamma_{\mathcal{R}},\\ \left\|\mathbf{e}_1^{\dagger}\right\| &\leq q_1/2 \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}} + \alpha_2^* \leq q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}.\end{aligned}$$

Therefore $\left\|(\mathbf{e}_0^{\dagger}, \mathbf{e}_1^{\dagger})\right\| \leq s \cdot q_0 \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}} \leq \beta_{q_2}^*$. This would, however, violate Assumption 4. We thus conclude that Case (i) is impossible.

If Case (ii) is true, we observe that

$$\begin{aligned}\left\|\mathbf{u}_0^{\dagger}\right\| &\leq s^2 \cdot t \cdot q_2 \cdot q_0 \cdot \beta \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}}^3 + t^3 \cdot q_2 \cdot q_0 \cdot \beta \cdot \alpha_3^* \cdot \gamma_{\mathcal{R}}^2\\ &\quad + s^2 \cdot q_2 \cdot q_1 \cdot \beta \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}}^3 + s^2 \cdot q_2 \cdot \beta \cdot \alpha_2^* \cdot \gamma_{\mathcal{R}}^2\\ &\leq (s+t)^3 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \beta \cdot \gamma_{\mathcal{R}}^3\\ &\leq \beta_{q_3}^*/2,\\ \left\|\mathbf{u}_0 - \mathbf{u}_0^{\dagger}\right\| &\leq \beta_{q_3}^*,\\ \left\|\mathbf{f}_0^{\mathsf{T}} \cdot \mathbf{e}_0^{\dagger} + \mathbf{f}_1^{\mathsf{T}} \cdot \mathbf{e}_1^{\dagger}\right\| &\leq (s+t) \cdot q_2 \cdot s \cdot q_0 \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^2\\ &\leq (s+t)^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^2\\ &\leq \beta_{q_3}^*.\end{aligned}$$

This would, however, violate Assumption 0. We thus conclude that Case (ii) is impossible.

Since both cases are impossible, we conclude that $(\mathbf{e}_0^{\dagger}, \mathbf{e}_1^{\dagger}) \neq \mathbf{0}$ with non-negligible probability. □

C.5.3 Efficiency - Proof of Theorem 4.8.3

Proof. Note that, $\log |\mathcal{R}_{q_3}| = \log q_3^{\varphi(\rho)} = O_\lambda(\log q_3) = O_\lambda(\log n)$, and an $\mathcal{R}_{q_3}$ operation takes at most $O_\lambda(\log^2 n)$ bit operations. The common reference string

$$\mathsf{crs} = \begin{pmatrix} \mathbf{D}_0, & \mathbf{t}_0, & (\mathbf{u}_{0,j})_{j \in I_0}, \\ \mathbf{D}_1, & \mathbf{t}_1, & (\mathbf{u}_{1,j})_{j \in I_1}, \\ \mathbf{D}_2, & \mathbf{t}_2, & (\mathbf{u}_{2,j})_{j \in I_2}, \\ \mathbf{D}_3, & \mathbf{t}_3, & (\mathbf{u}_{3,j})_{j \in I_3}, \\ v, & \mathbf{h}, & \mathbf{f}_0, \quad \mathbf{f}_1 \end{pmatrix}$$

has description size at most

$$(4 \cdot \eta \cdot (m+1) + 6 \cdot (s+t)) \cdot |\mathcal{R}_{q_3}| = O_\lambda(n \cdot \log n).$$

A proof $(c_{\mathbf{r}}, \mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ has description size at most

$$(4m+1) \cdot \log |\mathcal{R}_{q_3}| = O_\lambda(\log^2 n).$$

Preprocessing requires $O(n)$ $\mathcal{R}_{q_3}$ operations, which cost $O_\lambda(n \cdot \log^2 n)$ bit operations. After preprocessing, verification requires $O_\lambda(m)$ $\mathcal{R}_{q_3}$ operations, which cost $O_\lambda(\log^3 n)$ bit operations.

It remains to show that prover time is $O_\lambda(n \cdot \log^3 n)$. It suffices to analyse the time needed for computing $\mathbf{u}_{0,0}$ and $\mathbf{u}_{0,1}$ since they dominate the prover computation. Recall that

$$\mathbf{u}_{0,0} = \sum_{i\in[s],k\in[t]} f_{0,k} \cdot M_{k,i} \cdot \sum_{j\in[s]:j\neq i} \mathbf{u}_{0,j-i} \cdot x_j + \sum_{i,k\in[t]} f_{0,k} \cdot p \cdot \sum_{j\in[t]:j\neq i} \mathbf{u}_{0,j-i} \cdot r_j,$$
$$\mathbf{u}_{0,1} = \sum_{j\in[s]} h_j \cdot f_{1,j} \cdot \sum_{i\in[s]:i\neq j} \mathbf{u}_{0,i-j} \cdot x_i - \sum_{i\in[s]} f_{1,i} \cdot \sum_{j\in[s]:j\neq i} \mathbf{u}_{0,i-j} \cdot h_j \cdot x_j.$$

It is clear that once the terms

$$\sum_{j\in[s]:j\neq i} \mathbf{u}_{0,j-i} \cdot x_j, \qquad \sum_{j\in[t]:j\neq i} \mathbf{u}_{0,j-i} \cdot r_j,$$
$$\sum_{i\in[s]:i\neq j} \mathbf{u}_{0,i-j} \cdot x_i, \quad \text{and} \quad \sum_{j\in[s]:j\neq i} \mathbf{u}_{0,i-j} \cdot h_j \cdot x_j$$

are computed, $\mathbf{u}_{0,0}$ and $\mathbf{u}_{0,1}$ can be computed with $O_\lambda(n)$ $\mathcal{R}_{q_3}$ operations. We examine the cost for computing the first term, i.e. $\sum_{j\in[s]:j\neq i} \mathbf{u}_{0,j-i} \cdot x_j$.

Observe that $\sum_{j\in[s]:j\neq i} \mathbf{u}_{0,j-i} \cdot x_j$ can be written in the form

$$\begin{pmatrix} \mathbf{0} & \mathbf{u}_{0,1} & \mathbf{u}_{0,2} & \cdots & \cdots & \mathbf{u}_{0,s-1} \\ \mathbf{u}_{0,-1} & \mathbf{0} & \mathbf{u}_{0,1} & \ddots & & \vdots \\ \mathbf{u}_{0,-2} & \mathbf{u}_{0,-1} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{u}_{0,1} & \mathbf{u}_{0,2} \\ \vdots & & \ddots & \mathbf{u}_{0,-1} & \mathbf{0} & \mathbf{u}_{0,1} \\ \mathbf{u}_{0,-(s-1)} & \cdots & \cdots & \mathbf{u}_{0,-2} & \mathbf{u}_{0,-1} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_s \end{pmatrix}$$

which can be expressed as a sum of m matrix-vector products where each of the m matrices is an s-by-s Toeplitz matrix over $\mathcal{R}_{q_3}$. It is well-known (see e.g. [GL96]) that multiplying an s-by-s Toeplitz matrix to a vector takes $O(s \cdot \log s)$ operations over the base ring, i.e. $\mathcal{R}_{q_3}$. Therefore $\sum_{j\in[s]:j\neq i} \mathbf{u}_{0,j-i} \cdot x_j$ can be computed using $O_\lambda(n \cdot \log n \cdot m \cdot \log q) = O_\lambda(n \cdot \log^3 n)$ bit operations.

By splitting the other terms as sums of Toeplitz-vector products, we conclude that the computation of $\mathbf{u}_{0,0}$ and $\mathbf{u}_{0,1}$, and hence the the overall prover computation, takes time

$$O_\lambda(n \cdot \log^3 n).$$

□

C.6 Proofs for Applications

C.6.1 Completeness - Proof of Theorem 4.9.1

Proof. Since $\mathbf{x} \in \{0,1\}^s$, observe that

$$\|\mathbf{z}\| \leq \left\| \sum_{0 \leq i,j \leq s : i-j=k} h_j \cdot x_j \cdot (x_i - 1) \right\| \leq s \cdot \|\mathbf{h}\| .$$

For $\mathbf{h}$ generated by $\mathsf{Gen}^{\mathtt{str}}$, Lemma 4.6.6 implies that $\|\mathbf{h}\| \leq (q_1/2)^{\ell+1} \cdot \gamma_{\mathcal{R}}^{\ell}$. For $\mathbf{h}$ generated by $\mathsf{Gen}^{\mathtt{unstr}}$, we have $\mathbf{h} \in \mathcal{R}_{q_1}^s$ and thus $\|\mathbf{h}\| \leq q_1/2$. □

C.6.2 Knowledge-Soundness - Proof of Theorem 4.9.2

Proof. Fix a PPT prover $\mathcal{P}^*$ and let $\mathcal{P}_0^*$ and $\mathcal{P}_1^*$ be wrappers of $\mathcal{P}^*$ which interact with $\Pi'.\mathsf{Verify}$ and $\Pi''.\mathsf{Verify}$ respectively. By the knowledge-soundness of Π' and Π'', there exist knowledge extractors $\mathcal{E}_{\mathcal{P}_0^*}^{\Pi'}$ and $\mathcal{E}_{\mathcal{P}_1^*}^{\Pi''}$ respectively. Define an extractor $\mathcal{E}_{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}) = ((\mathbf{v}, \mathbf{h}), (\mathbf{M}, \mathbf{y}))$, does the following:

- Obtain $(c_{\mathbf{x}}, \bar{c}_{\mathbf{x}})$ from $\mathcal{P}^*$.
- Compute $c_{\mathbf{z}} \coloneqq \bar{c}_{\mathbf{x}} \cdot (c_{\mathbf{x}} - \langle \mathbf{v}, \mathbf{1} \rangle) \bmod q_3$.
- Obtain $\mathbf{x}^\dagger$ by running $\mathcal{E}_{\mathcal{P}_0^*}^{\Pi'}$ on $(\mathsf{crs}', ((\mathbf{M}, \mathbf{y}), (c_{\mathbf{x}}, \bar{c}_{\mathbf{x}}))$.
- If $\mathbf{x}^\dagger \in \{0,1\}[s]$, output $\mathbf{x}^\dagger$, else continue.
- Compute $\hat{z}_0 \coloneqq -\sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger$.
- If $\hat{z}_0 = 0$, output $((x_i^\dagger - 1) \cdot x_i^\dagger)_{i \in [s]}$, else continue.
- Obtain $\hat{\mathbf{z}}_{-0} = (\hat{z}_{-s}, \ldots, \hat{z}_{-1}, \hat{z}_1, \ldots, \hat{z}_s)$ by running $\mathcal{E}_{\mathcal{P}_1^*}^{\Pi''}$ on $(\mathsf{crs}'', (\epsilon, c_{\mathbf{z}}))$.
- Define $\hat{\mathbf{z}} \coloneqq (\hat{z}_{-s}, \ldots, \hat{z}_{-1}, \hat{z}_0, \hat{z}_1, \ldots, \hat{z}_s)$.
- Compute $\mathbf{z}^\dagger \coloneqq \left(\sum_{0 \leq i,j \leq s : i-j=k} h_j \cdot x_j^\dagger \cdot (x_i^\dagger - 1) \right)_{-s \leq k \leq s}$.
- Output $\hat{\mathbf{z}} - \mathbf{z}^\dagger$.

We claim that with overwhelming probability $\mathcal{E}_{\mathcal{P}^*}$ outputs $\mathbf{x}^\dagger$ such that $((\mathbf{M}, \mathbf{y}), \mathbf{x}^\dagger) \in \Psi^{\mathtt{str\text{-}bin\text{-}sat}}$ (resp. $\Psi^{\mathtt{bin\text{-}sat}}$).

First, by the knowledge-soundness of Π', we have with overwhelming probability that $\mathbf{M} \cdot \mathbf{x}^\dagger = \mathbf{y} \bmod q_0$ and that $\mathbf{x}^\dagger$ satisfies

$$\left\langle \mathbf{v}, \mathbf{x}^\dagger \right\rangle = c_{\mathbf{x}} \bmod q_3, \qquad \left\langle \bar{\mathbf{v}} \circ \mathbf{h}, \mathbf{x}^\dagger \right\rangle = \bar{c}_{\mathbf{x}} \bmod q_3, \qquad \left\| \mathbf{x}^\dagger \right\| \leq \alpha'.$$

It remains to argue that $\mathbf{x}^\dagger \in \{0,1\}^s$ with overwhelming probability.

Suppose towards a contradiction that $\mathbf{x}^\dagger \notin \{0,1\}^s$ with non-negligible probability. By the knowledge-soundness of Π'', with overwhelming probability $\hat{\mathbf{z}}_{-0}$ satisfies

$$\begin{aligned}
\langle (\bar{\mathbf{v}}||\mathbf{v}), \hat{\mathbf{z}}_{-0} \rangle &= c_{\mathbf{z}} \bmod q_3 \\
&= \bar{c}_{\mathbf{x}} \cdot (c_{\mathbf{x}} - \langle \mathbf{v}, \mathbf{1} \rangle) \bmod q_3 \\
&= \left(\sum_j \bar{v}_j \cdot h_j \cdot x_j^\dagger \right) \cdot \left(\sum_i v_i \cdot (x_i^\dagger - 1) \right) \bmod q_3 \\
&= \sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger + \sum_{i,j,i \neq j} v^{i-j} \cdot h_j \cdot (x_i^\dagger - 1) \cdot x_j^\dagger \bmod q_3, \\
\langle (\bar{\mathbf{v}}||1||\mathbf{v}), \hat{\mathbf{z}} \rangle &= \sum_{i,j,i \neq j} v^{i-j} \cdot h_j \cdot (x_i^\dagger - 1) \cdot x_j^\dagger \bmod q_3,
\end{aligned}$$

and $\|\hat{\mathbf{z}}\| \leq \alpha''$, where in the third equality we have used that the extracted vector $\mathbf{x}^\dagger$ satisfies $\left\langle \mathbf{v}, \mathbf{x}^\dagger \right\rangle = c_{\mathbf{x}} \bmod q_3$, and $\left\langle \bar{\mathbf{v}} \circ \mathbf{h}, \mathbf{x}^\dagger \right\rangle = \bar{c}_{\mathbf{x}} \bmod q_3$. On the other hand, $\mathbf{z}^\dagger$ satisfies

$$\left\langle (\bar{\mathbf{v}}||1||\mathbf{v}), \mathbf{z}^\dagger \right\rangle = \sum_{i,j,i \neq j} v^{i-j} \cdot h_j \cdot (x_i^\dagger - 1) \cdot x_j^\dagger \bmod q_3$$

with $z_0^\dagger = 0$ and $\left\|\mathbf{z}^\dagger\right\| \leq s \cdot \|\mathbf{h}\| \cdot (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}^2 \leq s \cdot (q_1/2)^{\ell+1} \cdot (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}^{\ell+2}$ (resp. $s \cdot q_1/2 \cdot (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}^2$). Therefore,

$$\left\langle (\bar{\mathbf{v}}||1||\mathbf{v}), \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\rangle = 0 \bmod q_3 \qquad \text{and} \qquad \left\|\hat{\mathbf{z}} - \mathbf{z}^\dagger\right\| \leq \beta_{q_3}.$$

One (or both) of the following two cases must be true

(i) $\sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger = 0$ with non-negligible probability.

(ii) $\sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger \neq 0$ with non-negligible probability,

If Case (i) is true, we have

$$\sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger = 0 \bmod q_1 \qquad \text{and} \qquad 0 < \left\|((x_i^\dagger - 1) \cdot x_i^\dagger)_{i \in [s]}\right\| \leq \beta_{q_1}$$

with non-negligible probability. This contradicts Assumption 0. If Case (ii) is true, we have

$$\left\langle (\bar{\mathbf{v}}||1||\mathbf{v}), \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\rangle = 0 \bmod q \qquad \text{and} \qquad 0 < \left\|\hat{\mathbf{z}} - \mathbf{z}^\dagger\right\| \leq \beta_{q_3}$$

with non-negligible probability. This contradicts Assumption 1. Since none of the two cases could be true, we must have $\mathbf{x}^\dagger \in \{0,1\}^s$, as claimed. □

C.6.3 Efficiency - Proof of Theorem 4.9.3

Proof. Note that $\log|\mathcal{R}_{q_3}| = \log q_3^{\varphi(\rho)} = O_\lambda(\log q_3) = O_\lambda(\log n)$, and an $\mathcal{R}_{q_3}$ operation takes at most $O_\lambda(\log^2 n)$ bit operations.

Notice that $\mathbf{z}$ can be computed in time $O_\lambda(n \cdot \log^3 n)$, exploiting fast multiplication algorithms for Toeplitz matrices (similarly to what described in Appendix C.5.3). All claims about the unstructured case then follow from Theorems 4.8.3 and 4.8.6.

For the structured case, we need to argue that crs has a short description size. Note that crs can be succinctly described by $(v, \tilde{\mathbf{h}}) \in \mathcal{R}_{q_3}^\times \times (\mathcal{R}_{q_3}^\times)^{\tilde{n}}$ where $\tilde{n} = \sum_{i=0}^{\ell-1}(k_i+1) + k_\ell$ and $n = \sum_{i=0}^{\ell} k_i$ with $k_i \in \{1,2\}$. We thus conclude that crs has description size $O_\lambda(\log^2 n)$. The rest of the claims for the structured case then follow from Theorems 4.7.3 and 4.7.6. □

C.7 Construction and Proofs for R1CS Argument

Let $\mathcal{R}, s_1, s_2, t, \eta, m, q_0, q_1, q_2, q_3, \alpha, \beta, \delta_0, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \mathcal{T}$ depend on λ. Using the lattice trapdoor algorithms (Section 4.3.2) parametrised by (η, m, q_3, β), in Figure C.2, we construct an argument system for $\Psi^{\mathtt{R1CS}}$.

C.7.1 Completeness

Theorem C.7.1 (Completeness)**.** *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 4.3.2 hold. For*

$$\delta_0 \geq 6 \cdot t^2 \cdot s^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^4, \qquad \delta_1 \geq s_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}},$$
$$\delta_2 \geq s \cdot t \cdot q_0 \cdot \cdot q_1 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3, \qquad \delta_3 \geq s \cdot t \cdot q_0 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2,$$
$$\delta_4 \geq s \cdot t \cdot q_0 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2, \qquad \delta_5 \geq s^2 \cdot t \cdot q_0 \cdot \alpha^2 \cdot \beta \cdot \gamma_{\mathcal{R}}^3,$$
$$\delta_6 \geq 3 \cdot s^2 \cdot t^2 \cdot q_0^2 \cdot q_1^2 \cdot \alpha^2 \cdot \beta \cdot \gamma_{\mathcal{R}}^3,$$

$\Pi^{\mathtt{R1CS}}$ in Figure C.2 is complete.

The proof of this claim is essentially identical to that of Theorem 4.8.1, and is therefore omitted.

C.7.2 Knowledge Soundness

Theorem C.7.2 (Knowledge Soundness)**.** *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 4.3.2 hold. Let $w = 1$, $\mathcal{G} := \{X^j : -s \leq j \leq s\}$, $\mathcal{G}_0 = \{X^i : i \in \pm[\max\{s,t\}]\}$, $\mathcal{G}_1 = \{X^i : i \in [s_1; s]\}$, $\mathcal{G}_2 = \{X^i : i \in -[t]\}$, $\mathcal{G}_3 = \{X^i : i \in [t]\}$, $\mathcal{G}_4 = \{X^i : i \in [t]\}$, $\mathcal{G}_5 = \{X^i : i \in [t]\}$, and $\mathcal{G}_6 = \{X^i : i \in \pm[t]\}$ be sets of monomials in X. Let $\mathcal{D}$ denote the distribution $\mathsf{SampD}(1^\lambda)$. For $i \in$*

$\mathsf{Setup}(1^\lambda, \mathsf{pp})$

$(v, \mathbf{h}, \boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \boldsymbol{\ell}_3) \leftarrow \mathcal{R}_{q_3}^{\times} \times \mathcal{R}_{q_1}^{t} \times \mathcal{R}_{q_2}^{t} \times \mathcal{R}_{q_2}^{t} \times \mathcal{R}_{q_2}^{t}$
$I_0 \coloneqq \pm[\max\{s, t\}],\ I_1 \coloneqq [s_1 + 1; s],$
$I_2 \coloneqq -[t],\ I_3 \coloneqq [t],\ I_4 \coloneqq [t],\ I_5 \coloneqq [t]$
$I_6 \coloneqq \pm[t]$
for $i \in \{0, 1, 2, 3, 4, 5, 6\}$ **do**
$(\mathbf{D}_i, \mathsf{td}_i) \leftarrow \mathsf{TrapGen}(1^\lambda), \quad \mathbf{t}_i \leftarrow \mathcal{T}$
$\mathbf{u}_{i,j} \leftarrow \mathsf{SampPre}(\mathsf{td}_i, \mathbf{t}_i \cdot v^j),\ \forall j \in I_i$

$$\mathsf{crs} \coloneqq \begin{pmatrix} (\mathbf{D}_i, & \mathbf{t}_i, & (\mathbf{u}_{i,j})_{j \in I_i})_{i=0}^{6}, \\ v, & \mathbf{h}, & \boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \boldsymbol{\ell}_3 \end{pmatrix}$$

return crs

$\mathsf{PreVerify}(\mathsf{crs}, (\mathbf{x}_1, \mathbf{E}, \mathbf{F}, \mathbf{G}))$

$c_{\mathbf{x}_1} \coloneqq \mathbf{v}_1^\mathsf{T} \cdot \mathbf{x}_1 \bmod q_3$
$\bar{c}_{\mathbf{E}} \coloneqq (\boldsymbol{\ell}_1 \cdot \mathbf{h})^\mathsf{T} \cdot \mathbf{E} \cdot \bar{\mathbf{v}} \bmod q_3$
$\bar{c}_{\mathbf{F}} \coloneqq \boldsymbol{\ell}_2^\mathsf{T} \cdot \mathbf{F} \cdot \bar{\mathbf{v}} \bmod q_3$
$\bar{c}_{\mathbf{G}} \coloneqq \boldsymbol{\ell}_3^\mathsf{T} \cdot \mathbf{G} \cdot \bar{\mathbf{v}} \bmod q_3$
$c_{\mathbf{I},1} \coloneqq \mathbf{v}_t^\mathsf{T} \cdot \boldsymbol{\ell}_1 \bmod q_3$
$\bar{c}_{\mathbf{I},2} \coloneqq \bar{\mathbf{v}}_t^\mathsf{T} \cdot \boldsymbol{\ell}_2 \bmod q_3$
$\bar{c}_{\mathbf{I},3} \coloneqq \bar{\mathbf{v}}_t^\mathsf{T} \cdot \boldsymbol{\ell}_3 \bmod q_3$
$\bar{c}_{\mathbf{I},6} \coloneqq \bar{\mathbf{v}}_t^\mathsf{T} \cdot \mathbf{h} \bmod q_3$

$$\mathsf{crs}_{\mathbf{E},\mathbf{F},\mathbf{G}} \coloneqq \begin{pmatrix} (\mathbf{D}_i, \mathbf{t}_i)_{i=0}^{6}, \\ c_{\mathbf{x}_1}, \bar{c}_{\mathbf{E}}, \bar{c}_{\mathbf{F}}, \bar{c}_{\mathbf{G}}, \\ c_{\mathbf{I},1}, \bar{c}_{\mathbf{I},2}, \bar{c}_{\mathbf{I},3}, \bar{c}_{\mathbf{I},6} \end{pmatrix}$$

return $\mathsf{crs}_{\mathbf{x}_1,\mathbf{E},\mathbf{F},\mathbf{G}}$

$\mathsf{Verify}(\mathsf{crs}_{\mathbf{x}_1,\mathbf{E},\mathbf{F},\mathbf{G}}, \pi)$

$c_{\mathbf{x}} \coloneqq c_{\mathbf{x}_1} + c_{\mathbf{x}_2} \bmod q_3$
$c_{0,\mathbf{E}} \coloneqq \bar{c}_{\mathbf{E}} \cdot c_{\mathbf{x}} - c_{\mathbf{I},1} \cdot \bar{c}_{\mathbf{e}} \bmod q_3$
$c_{0,\mathbf{F}} \coloneqq \bar{c}_{\mathbf{F}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{I},2} \cdot c_{\mathbf{f}} \bmod q_3$
$c_{0,\mathbf{G}} \coloneqq \bar{c}_{\mathbf{G}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{I},3} \cdot c_{\mathbf{g}} \bmod q_3$
$c_0 \coloneqq c_{0,\mathbf{E}} + c_{0,\mathbf{F}} + c_{0,\mathbf{G}} \bmod q_3$
$(c_1, c_2, c_3, c_4, c_5) \coloneqq (c_{\mathbf{x}_2}, \bar{c}_{\mathbf{e}}, c_{\mathbf{f}}, c_{\mathbf{g}}, c_{\mathbf{r}})$
$c_6 \coloneqq \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{r}} - \bar{c}_{\mathbf{I},\mathbf{6}} \cdot c_{\mathbf{g}} \bmod q_3$
for $i \in \{0, 1, 2, 3, 4, 5, 6\}$ **do**
$b_i \coloneqq (\mathbf{D}_i \cdot \mathbf{u}_i \overset{?}{\equiv} \mathbf{t}_i \cdot c_i \bmod q_3 \ \wedge\ \|\mathbf{u}_i\| \overset{?}{\leq} \delta_i)$
return $b_0 \wedge b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge b_5 \wedge b_6$

$\mathsf{Prove}(\mathsf{crs}, (\mathbf{E}, \mathbf{F}, \mathbf{G}), \mathbf{x}_2)$

$c_{\mathbf{x}_2} \coloneqq \mathbf{v}_2^\mathsf{T} \cdot \mathbf{x}_2 \bmod q_3$
$c_{\mathbf{r}} \coloneqq \mathbf{v}_t^\mathsf{T} \cdot \mathbf{r} \bmod q_3$
$\bar{c}_{\mathbf{e}} \coloneqq (\bar{\mathbf{v}}_t \circ \mathbf{h})^\mathsf{T} \cdot \mathbf{E} \cdot \mathbf{x} \bmod q_3$
$c_{\mathbf{f}} \coloneqq \mathbf{v}_t^\mathsf{T} \cdot \mathbf{F} \cdot \mathbf{x} \bmod q_3$
$c_{\mathbf{g}} \coloneqq \mathbf{v}_t^\mathsf{T} \cdot \mathbf{G} \cdot \mathbf{x} \bmod q_3$

$$\mathbf{u}_{\mathbf{E}} \coloneqq \sum_{i \in [t], j \in [s]} E_{i,j} \cdot h_i \cdot \ell_{1,i} \cdot \sum_{k \in [s]: k \neq j} \mathbf{u}_{0,k-j} \cdot x_k + \sum_{i \in [t], j \in [s]} E_{i,j} \cdot x_j \cdot h_i \cdot \sum_{k \in [t]: k \neq i} \mathbf{u}_{0,k-i} \cdot \ell_{1,k}$$

$$\mathbf{u}_{\mathbf{F}} \coloneqq \sum_{i \in [t], j \in [s]} F_{i,j} \cdot \ell_{2,i} \cdot \sum_{k \in [s]: k \neq j} \mathbf{u}_{0,k-j} \cdot x_k + \sum_{i \in [t], j \in [s]} F_{i,j} \cdot x_j \cdot \sum_{k \in [t]: k \neq i} \mathbf{u}_{0,i-k} \cdot \ell_{2,k}$$

$$\mathbf{u}_{\mathbf{G}} \coloneqq \sum_{i \in [t], j \in [s]} G_{i,j} \cdot \ell_{3,i} \cdot \sum_{k \in [s]: k \neq j} \mathbf{u}_{0,k-j} \cdot x_k + \sum_{i \in [t], j \in [s]} G_{i,j} \cdot x_j \cdot \sum_{k \in [t]: k \neq i} \mathbf{u}_{0,i-k} \cdot \ell_{3,k}$$

$\mathbf{u}_0 \coloneqq \mathbf{u}_{\mathbf{E}} + \mathbf{u}_{\mathbf{F}} + \mathbf{u}_{\mathbf{G}}$

$$\mathbf{u}_1 \coloneqq \sum_{j \in [s_1+1; s]} \mathbf{u}_{1,j} \cdot x_j$$

$$\mathbf{u}_2 \coloneqq \sum_{i \in [t]} \sum_{j \in [s]} \mathbf{u}_{2,-i} \cdot E_{i,j} \cdot h_i \cdot x_j$$

$$\mathbf{u}_3 \coloneqq \sum_{i \in [t]} \sum_{j \in [s]} \mathbf{u}_{3,i} \cdot F_{i,j} \cdot x_j$$

$$\mathbf{u}_4 \coloneqq \sum_{i \in [t]} \sum_{j \in [s]} \mathbf{u}_{4,i} \cdot G_{i,j} \cdot x_j$$

$$\mathbf{u}_5 \coloneqq \sum_{i \in [t]} \mathbf{u}_{5,i} \cdot r_i$$

$$\mathbf{u}_6 \coloneqq \sum_{i,j \in [t], i \neq j} \mathbf{u}_{6,i-j} \cdot (e_j \cdot f_i + q_0 \cdot r_i - g_i) \cdot h_j$$

$$\textbf{return } \pi \coloneqq \begin{pmatrix} c_{\mathbf{x}_2}, c_{\mathbf{r}}, \bar{c}_{\mathbf{e}}, c_{\mathbf{f}}, c_{\mathbf{g}}, \\ \mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4, \mathbf{u}_5, \mathbf{u}_6 \end{pmatrix}$$

Figure C.2: Our argument system Π^{R1CS}.

$\{1,2,3,4,5,6\}$, *let* $\mathcal{Z}_i(1^\lambda)$ *be almost identical to* $\mathsf{Setup}(1^\lambda, \mathsf{Gen}^{\mathtt{unstr}}(1^\lambda))$, *except that it is given* $(\mathbf{D}_i, \mathbf{t}_i, v, \{\mathbf{u}_{i,j}\}_{j\in I_i})$ *as input and generates the rest of* crs. *Let*

$$\begin{aligned}
\alpha_i^* &\geq \delta_i, \ \forall i \in [6] \\
\alpha^* &:= \max\{\alpha_1^*, \alpha_2^*, \alpha_3^*, \alpha_4^*, \alpha_5^*, \alpha_6^*\}, \\
q_1 \geq \beta_{q_1}^* &\geq s \cdot q_0 \cdot (\alpha^*)^2 \cdot \gamma_{\mathcal{R}} \\
q_2 \geq \beta_{q_2}^* &\geq t \cdot s \cdot q \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^2, \\
q_3 \geq \beta_{q_3} &\geq t \cdot s \cdot q_0 \cdot q_1 \cdot (\alpha^*)^2 \cdot \gamma_{\mathcal{R}}^3, \\
q_3 \geq \beta_{q_3}^* &\geq \max\{2\delta_0, (s+t)^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \beta \cdot \gamma_{\mathcal{R}}^4\}.
\end{aligned}$$

$\Pi^{\mathtt{R1CS}}$ *in Figure C.2 is knowledge-sound for* $\Psi^{\mathtt{R1CS}}[\alpha_1^*]$ *if the following assumptions hold:*

Assumption 0. k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\beta,\beta_{q_3}^*,\mathcal{G}_0,g^*=1,\mathcal{D},\mathcal{T}}$,

Assumption 1. *knowledge-*k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_1^*,\beta,\delta_1,\mathcal{G}_1,\mathcal{D},\mathcal{T},\mathcal{Z}_1}$,

Assumption 2. *knowledge-*k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_2^*,\beta,\delta_2,\mathcal{G}_2,\mathcal{D},\mathcal{T},\mathcal{Z}_2}$,

Assumption 3. *knowledge-*k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_3^*,\beta,\delta_3,\mathcal{G}_3,\mathcal{D},\mathcal{T},\mathcal{Z}_3}$,

Assumption 4. *knowledge-*k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_4^*,\beta,\delta_4,\mathcal{G}_4,\mathcal{D},\mathcal{T},\mathcal{Z}_4}$,

Assumption 5. *knowledge-*k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_5^*,\beta,\delta_5,\mathcal{G}_5,\mathcal{D},\mathcal{T},\mathcal{Z}_5}$,

Assumption 6. *knowledge-*k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_6^*,\beta,\delta_6,\mathcal{G}_6,\mathcal{D},\mathcal{T},\mathcal{Z}_6}$,

Assumption 7. R-$\mathsf{SIS}_{\mathcal{R},t,q_1,\beta_{q_1}^*}$,

Assumption 8. R-$\mathsf{SIS}_{\mathcal{R},3\cdot t,q_2,\beta_{q_2}^*}$, *and*

Assumption 9. $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},1,q_3,\beta_{q_3}}$.

Proof. Fix a PPT prover $\mathcal{P}^*$. Consider an algorithm $\mathcal{B}_1 = \mathcal{B}^{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, runs $\pi \leftarrow \mathcal{P}^*(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, parses $c_{\mathbf{x}_2}$ and $\mathbf{u}_1$ from π, and outputs $(c_{\mathbf{x}_2}, \mathbf{u}_1)$. Similarly, consider the algorithms $\mathcal{B}_2 = \mathcal{B}^{\mathcal{P}^*}$, $\mathcal{B}_3 = \mathcal{B}^{\mathcal{P}^*}$, $\mathcal{B}_4 = \mathcal{B}^{\mathcal{P}^*}$, $\mathcal{B}_5 = \mathcal{B}^{\mathcal{P}^*}$, and $\mathcal{B}_6 = \mathcal{B}^{\mathcal{P}^*}$ which do almost the same, except that

- $\mathcal{B}_2 = \mathcal{B}^{\mathcal{P}^*}$ extracts $(\bar{c}_{\mathbf{e}}, \mathbf{u}_2)$ from π ,
- $\mathcal{B}_3 = \mathcal{B}^{\mathcal{P}^*}$ extracts $(c_{\mathbf{f}}, \mathbf{u}_3)$ from π,
- $\mathcal{B}_4 = \mathcal{B}^{\mathcal{P}^*}$ extracts $(c_{\mathbf{g}}, \mathbf{u}_4)$ from π,
- $\mathcal{B}_5 = \mathcal{B}^{\mathcal{P}^*}$ extracts $(c_{\mathbf{r}}, \mathbf{u}_5)$ from π, and

- $\mathcal{B}_6 = \mathcal{B}^{\mathcal{P}^*}$ parses $(\bar{c}_{\mathbf{e}}, c_{\mathbf{f}}, c_{\mathbf{r}}, c_{\mathbf{g}}, \mathbf{u}_6)$ from π, computes $\bar{c}_{\mathbf{I},6} := \bar{\mathbf{v}}_t^{\mathsf{T}} \cdot \mathbf{h}$ and

$$c_{\mathbf{z}} := \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{r}} - \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{g}},$$

  and outputs $(c_{\mathbf{z}}, \mathbf{u}_6)$.

Let $\mathcal{E}_{\mathcal{B}_1}^{k\text{-}R\text{-}\mathsf{ISIS},1}$, $\mathcal{E}_{\mathcal{B}_2}^{k\text{-}R\text{-}\mathsf{ISIS},2}$, $\mathcal{E}_{\mathcal{B}_3}^{k\text{-}R\text{-}\mathsf{ISIS},3}$, $\mathcal{E}_{\mathcal{B}_4}^{k\text{-}R\text{-}\mathsf{ISIS},4}$, $\mathcal{E}_{\mathcal{B}_5}^{k\text{-}R\text{-}\mathsf{ISIS},5}$, and $\mathcal{E}_{\mathcal{B}_6}^{k\text{-}R\text{-}\mathsf{ISIS},6}$ be the knowledge extractors whose existence are guaranteed by Assumptions 1, 2, 3, 4, 5, and 6. Define an extractor $\mathcal{E}_{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, does the following:

- run $\mathbf{x}_2^\dagger \leftarrow \mathcal{E}_{\mathcal{B},1}^{k\text{-}R\text{-}\mathsf{ISIS},1}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{e}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},2}^{k\text{-}R\text{-}\mathsf{ISIS},2}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{f}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},3}^{k\text{-}R\text{-}\mathsf{ISIS},3}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{g}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},4}^{k\text{-}R\text{-}\mathsf{ISIS},4}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{r}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},5}^{k\text{-}R\text{-}\mathsf{ISIS},5}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{z}_{-0}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},6}^{k\text{-}R\text{-}\mathsf{ISIS},6}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- check that $\mathbf{e}^\dagger = \mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$,
- check that $\mathbf{f}^\dagger = \mathbf{F} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$
- check that $\mathbf{g}^\dagger = \mathbf{G} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$
- check that $\left(\mathbf{E} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right) \circ \left(\mathbf{F} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right) + q_0 \cdot \mathbf{r}^\dagger = \left(\mathbf{G} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right)$, and
- output $\mathbf{x}_2^\dagger$ if all checks pass.

□

Fix any adversary $\mathcal{A}$ and consider the following experiment Exp:

$\mathsf{Exp}(1^\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Gen}^{\mathtt{unstr}}(1^\lambda)$
$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp})$
$(\mathsf{stmt}, \mathsf{wit}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{crs})$
$(\pi, \mathsf{wit}^\dagger) \leftarrow (\mathcal{P}^* | \mathcal{E}_{\mathcal{P}^*})(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$
$\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$
return $\mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}, \pi) = 1 \wedge (\mathsf{stmt}, \mathsf{wit}^\dagger) \notin \Psi_{\mathsf{pp}}$

We claim that $\Pr[\mathsf{Exp}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$, which proves the theorem.

To prove the claim, consider a modified experiment Exp' where in the setup $\mathsf{Setup}(1^\lambda, \mathsf{pp})$ the matrices $(\mathbf{D}_i)_{i=0}^6$ are sampled uniformly at random and the $\mathsf{SampPre}$ steps are replaced with sampling from SampD subject to the appropriate constraints. By the properties of $(\mathsf{TrapGen}, \mathsf{SampD}, \mathsf{SampPre})$, Exp' is statistically close to Exp. Therefore it suffices to show that $\Pr[\mathsf{Exp}'(1^\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

We now examine $\mathsf{wit}^\dagger$ generated during the execution of $\mathsf{Exp}'(1^\lambda)$. Parse $\mathsf{stmt} = (\mathbf{x}_1, (\mathbf{E}, \mathbf{F}, \mathbf{G}))$ and $\mathsf{wit}^\dagger = \mathbf{x}_2^\dagger$. First, suppose that $\mathcal{E}_{\mathcal{P}^*}$ returns something, i.e.

1. $\mathbf{e}^\dagger = \mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$,
2. $\mathbf{f}^\dagger = \mathbf{F} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$,
3. $\mathbf{g}^\dagger = \mathbf{G} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$, and
4. $\left(\mathbf{E} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right) \circ \left(\mathbf{F} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right) + q_0 \cdot \mathbf{r}^\dagger = \left(\mathbf{G} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right)$,

then by Conditions b_1, b_2, b_3, b_4, and b_5 of the verification algorithm and Assumptions 1, 2, 3, 4, and 5, we have

$$\begin{aligned} c_{\mathbf{x}_2} &= \mathbf{v}_2^\mathsf{T} \cdot \mathbf{x}_2^\dagger \bmod q_3, & \left\|\mathbf{x}_2^\dagger\right\| &\leq \alpha_1^* \\ \bar{c}_{\mathbf{e}} &= \bar{\mathbf{v}}_t^\mathsf{T} \cdot \mathbf{e}^\dagger \bmod q_3, & \left\|\mathbf{e}^\dagger\right\| &\leq \alpha_2^* \\ c_{\mathbf{f}} &= \mathbf{v}_t^\mathsf{T} \cdot \mathbf{f}^\dagger \bmod q_3, & \left\|\mathbf{f}^\dagger\right\| &\leq \alpha_3^* \\ c_{\mathbf{g}} &= \mathbf{v}_t^\mathsf{T} \cdot \mathbf{g}^\dagger \bmod q_3, & \left\|\mathbf{g}^\dagger\right\| &\leq \alpha_4^* \\ c_{\mathbf{r}} &= \mathbf{v}_t^\mathsf{T} \cdot \mathbf{r}^\dagger \bmod q_3, \quad \text{and} & \left\|\mathbf{r}^\dagger\right\| &\leq \alpha_5^*, \end{aligned}$$

Let us first show that Item 1, Item 2, and Item 3 hold.

Let $\mathbf{x}^\dagger = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$. Examining the condition b_0 in the verification algorithm, we observe

$$\begin{aligned}
\mathbf{D}_0 \cdot \mathbf{u}_{0,\mathbf{E}} &= \mathbf{t}_0 \cdot (\bar{c}_{\mathbf{E}} \cdot c_{\mathbf{x}} - c_{\mathbf{I},1} \cdot \bar{c}_{\mathbf{e}}) \\
&= \mathbf{t}_0 \cdot ((\boldsymbol{\ell}_1 \circ \mathbf{h})^\mathsf{T} \cdot \mathbf{E} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x}^\dagger - \mathbf{v}_t^\mathsf{T} \cdot \boldsymbol{\ell}_1 \cdot \bar{\mathbf{v}}_t^\mathsf{T} \cdot \mathbf{e}^\dagger) \\
&= \mathbf{t}_0 \cdot \boldsymbol{\ell}_1^\mathsf{T} \cdot (\mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} - \mathbf{I}_s) \cdot \mathbf{x}^\dagger - (\mathbf{v}_t \cdot \bar{\mathbf{v}}_t^\dagger - \mathbf{I}_t) \cdot \mathbf{e}^\dagger + \mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot \mathbf{x}^\dagger - \mathbf{e}^\dagger) \bmod q_3 \\
\mathbf{D}_0 \cdot \mathbf{u}_{0,\mathbf{F}} &= \mathbf{t}_0 \cdot (\bar{c}_{\mathbf{F}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{I},2} \cdot c_{\mathbf{f}}) \\
&= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_2^\mathsf{T} \cdot \mathbf{F} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x}^\dagger - \bar{\mathbf{v}}_t^\mathsf{T} \cdot \boldsymbol{\ell}_2 \cdot \mathbf{v}_t^\mathsf{T} \cdot \mathbf{f}^\dagger) \\
&= \mathbf{t}_0 \cdot \boldsymbol{\ell}_2^\mathsf{T} \cdot (\mathbf{F} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} - \mathbf{I}_s) \cdot \mathbf{x}^\dagger - (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\mathsf{T} - \mathbf{I}_t) \cdot \mathbf{f}^\dagger + \mathbf{F} \cdot \mathbf{x}^\dagger - \mathbf{f}^\dagger) \bmod q_3, \\
\mathbf{D}_0 \cdot \mathbf{u}_{0,\mathbf{G}} &= \mathbf{t}_0 \cdot (\bar{c}_{\mathbf{G}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{I},3} \cdot c_{\mathbf{g}}) \\
&= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_3^\mathsf{T} \cdot \mathbf{G} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x}^\dagger - \bar{\mathbf{v}}_t^\mathsf{T} \cdot \boldsymbol{\ell}_3 \cdot \mathbf{v}_t^\mathsf{T} \cdot \mathbf{g}^\dagger) \\
&= \mathbf{t}_0 \cdot \boldsymbol{\ell}_3^\mathsf{T} \cdot (\mathbf{G} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} - \mathbf{I}_s) \cdot \mathbf{x}^\dagger - (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\mathsf{T} - \mathbf{I}_t) \cdot \mathbf{g}^\dagger + \mathbf{G} \cdot \mathbf{x}^\dagger - \mathbf{g}^\dagger) \bmod q_3.
\end{aligned}$$

Let

$$\begin{aligned}
\mathbf{u}_{0,\mathbf{E}}^\dagger &\coloneqq \sum_{i\in[t],j,k\in[s]:k\neq j} \ell_{1,i} \cdot h_i \cdot E_{i,j} \cdot \mathbf{u}_{0,k-j} \cdot x_k^\dagger + \sum_{i,k\in[t]:k\neq i} \ell_{1,k} \cdot \mathbf{u}_{0,k-i} \cdot e_i^\dagger \\
\mathbf{u}_{0,\mathbf{F}}^\dagger &\coloneqq \sum_{i\in[t],j,k\in[s]:k\neq j} \ell_{2,i} \cdot F_{i,j} \cdot \mathbf{u}_{0,k-j} \cdot x_k^\dagger + \sum_{i,k\in[t]:k\neq i} \ell_{2,i} \cdot \mathbf{u}_{0,i-k} \cdot f_k^\dagger \\
\mathbf{u}_{0,\mathbf{G}}^\dagger &\coloneqq \sum_{i\in[t],j,k\in[s]:k\neq j} \ell_{3,i} \cdot G_{i,j} \cdot \mathbf{u}_{0,k-j} \cdot x_k^\dagger + \sum_{i,k\in[t],j\in[s]:i\neq j} G_{i,j} \cdot x_j^\dagger \cdot \ell_{3,i} \cdot \mathbf{u}_{0,i-k} \cdot h_k,
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{w}_1^\dagger &\coloneqq \mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot \mathbf{x}^\dagger - \mathbf{e}^\dagger \\
\mathbf{w}_2^\dagger &\coloneqq \mathbf{F} \cdot \mathbf{x}^\dagger - \mathbf{f}^\dagger \\
\mathbf{w}_3^\dagger &\coloneqq \mathbf{G} \cdot \mathbf{x}^\dagger - \mathbf{g}^\dagger
\end{aligned}$$

We have

$$\begin{aligned}
\mathbf{D}_0 \cdot (\mathbf{u}_{0,\mathbf{E}} - \mathbf{u}_{0,\mathbf{E}}^\dagger) &= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_1^\mathsf{T} \cdot \mathbf{w}_1^\dagger) \bmod q_3 \\
\mathbf{D}_0 \cdot (\mathbf{u}_{0,\mathbf{F}} - \mathbf{u}_{0,\mathbf{F}}^\dagger) &= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_2^\mathsf{T} \cdot \mathbf{w}_2^\dagger) \bmod q_3 \\
\mathbf{D}_0 \cdot (\mathbf{u}_{0,\mathbf{G}} - \mathbf{u}_{0,\mathbf{G}}^\dagger) &= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_3^\mathsf{T} \cdot \mathbf{w}_3^\dagger) \bmod q_3.
\end{aligned}$$

Suppose, contrary to our claim, that $\mathbf{w}^\dagger \coloneqq (\mathbf{w}_1^\dagger, \mathbf{w}_2^\dagger, \mathbf{w}_3^\dagger) \neq \mathbf{0}$ with non-negligible probability. Then, one (or both) of the following must be true:

(1) $\boldsymbol{\ell}^{\mathsf{T}} \cdot \mathbf{w}^{\dagger} = \mathbf{0}$ with non-negligible probability

(2) $\boldsymbol{\ell}^{\mathsf{T}} \cdot \mathbf{w}^{\dagger} \neq \mathbf{0}$ with non-negligible probability,

where $\boldsymbol{\ell} \coloneqq (\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \boldsymbol{\ell}_3)$. If Case (1) is true, then we also have with non-negligible probability

$$\boldsymbol{\ell}^{\mathsf{T}} \cdot \mathbf{w}^{\dagger} = \mathbf{0} \bmod q_2.$$

Note that

$$\begin{aligned}
\left\|\mathbf{w}_1^{\dagger}\right\| &\leq t \cdot s \cdot q_1/2 \cdot q_0/2 \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}}^2 + \alpha_2^* \leq t \cdot s \cdot q_0 \cdot q_1 \cdot \gamma_{\mathcal{R}}^2 \cdot \alpha^* \\
\left\|\mathbf{w}_2^{\dagger}\right\| &\leq s \cdot q_0/2 \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}} + \alpha_3^* \leq s \cdot q_0 \cdot \gamma_{\mathcal{R}} \cdot \alpha^* \\
\left\|\mathbf{w}_3^{\dagger}\right\| &\leq s \cdot q_0/2 \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}} + \alpha_3^* \leq s \cdot q_0 \cdot \gamma_{\mathcal{R}} \cdot \alpha^*,
\end{aligned}$$

Therefore $\left\|\mathbf{w}^{\dagger}\right\| \leq t \cdot s \cdot q_0 \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^2 \leq \beta_{q_2}^*$. This would, however, violate Assumption 8. We thus conclude that Case (1) is impossible.

If Case (2) is true, we observe that for each $j \in \{\mathbf{E}, \mathbf{F}, \mathbf{G}\}$

$$\begin{aligned}
\left\|\mathbf{u}_{0,j}^{\dagger}\right\| &\leq 2 \cdot t^2 \cdot s \cdot q_0/2 \cdot q_1/2 \cdot q_2/2 \cdot \beta \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^4 \\
&\leq (s+t)^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \beta \cdot \gamma_{\mathcal{R}}^4 \\
&\leq \beta_{q_3}^*/6, \\
\left\|\mathbf{u}_{0,j} - \mathbf{u}_{0,j}^{\dagger}\right\| &\leq \beta_{q_3}^*/3
\end{aligned}$$

Therefore

$$\left\|\mathbf{u}_{0,\mathbf{E}} - \mathbf{u}_{0,\mathbf{E}}^{\dagger} + \mathbf{u}_{0,\mathbf{F}} - \mathbf{u}_{0,\mathbf{F}}^{\dagger} + \mathbf{u}_{0,\mathbf{G}} - \mathbf{u}_{0,\mathbf{G}}^{\dagger}\right\| \leq \beta_{q_3}^*.$$

Moreover

$$\left\|\boldsymbol{\ell}^{\mathsf{T}} \cdot \mathbf{w}^{\dagger}\right\| \leq (t+s)^2 \cdot q_2 \cdot s \cdot q_0 \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}} \leq \beta_{q_3}^*.$$

This would, however, violate Assumption 0. We thus conclude that Case (2) is impossible.

It remains to show that Item 4 also holds, so that $\mathcal{E}_{\mathcal{P}^*}$ returns something with overwhelming probability. Suppose, for the sake of contradiction, that this is not the case. Let $\hat{\mathbf{e}} \coloneqq \mathbf{E} \cdot \mathbf{x}^{\dagger}$, i.e., $\mathbf{e}^{\dagger} = \mathsf{diag}(\mathbf{h}) \cdot \hat{\mathbf{e}}$. Compute $z_0^{\dagger} \coloneqq -\sum_{i \in [t]} (\hat{e}_i \cdot f_i^{\dagger} + q_0 \cdot r_i^{\dagger} + g_i^{\dagger}) \cdot h_i$. Then

$$\begin{aligned}
\left\|(\hat{e}_i \cdot f_i^{\dagger} + q_0 \cdot r_i^{\dagger} + g_i^{\dagger})_i\right\| &\leq s \cdot q_0/2 \cdot \alpha_1^* \cdot \alpha_3^* \cdot \gamma_{\mathcal{R}} + q_0 \cdot \alpha_5^* + \alpha_4^* \\
&\leq s \cdot q_0 \cdot (\alpha^*)^2 \cdot \gamma_{\mathcal{R}} \\
&\leq \beta_{q_1}^*
\end{aligned}$$

By Condition b_6 of the verification algorithm and Assumption 6, we have

$$\begin{aligned}
(\bar{\mathbf{v}}||\mathbf{v})^{\mathsf{T}} \cdot \mathbf{z}_{-0}^{\dagger} &= c_{\mathbf{z}} \bmod q_3 \\
&= \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{r}} - \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{g}} \bmod q_3 \\
&= \left(\sum_{j\in[t]} v^{-j} \cdot e_j^{\dagger}\right) \cdot \left(\sum_{i\in[t]} v^{i} \cdot f_i^{\dagger}\right) + q_0 \cdot \left(\sum_{j\in[t]} v^{-j} \cdot h_j\right) \cdot \left(\sum_{i\in[t]} v^{i} \cdot r_i^{\dagger}\right) \\
&\qquad - \left(\sum_{i\in[t]} g_i^{\dagger} \cdot v^{i}\right) \cdot \left(\sum_{j\in[t]} h_j \cdot v^{-j}\right) \\
&= \sum_{i,j\in[t]} \hat{e}_j \cdot f_i^{\dagger} \cdot h_j \cdot v^{i-j} + q_0 \cdot \sum_{i,j\in[t]} r_i^{\dagger} \cdot h_j \cdot v^{i-j} - \sum_{i,j\in[t]} g_i^{\dagger} \cdot h_j \cdot v^{i-j} \\
&= \sum_{i,j\in[t]} \left(\hat{e}_j \cdot f_i^{\dagger} + q_0 \cdot r_i^{\dagger} - g_i^{\dagger}\right) \cdot h_j \cdot v^{i-j}.
\end{aligned}$$

If $\mathbf{z}_{-0}^{\dagger} = (z_{-s}^{\dagger}, \ldots, z_{-1}^{\dagger}, z_1^{\dagger}, \ldots, z_s^{\dagger})$, and we let $\mathbf{z}^{\dagger} \coloneqq (z_{-s}^{\dagger}, \ldots, z_{-1}^{\dagger}, z_0^{\dagger}, z_1^{\dagger}, \ldots, z_s^{\dagger})$, we obtain

$$(\bar{\mathbf{v}}||1||\mathbf{v})^{\mathsf{T}} \cdot \mathbf{z}^{\dagger} = \sum_{i,j\in[t],i\neq j} (\hat{e}_j \cdot f_i^{\dagger} + q_0 \cdot r_i^{\dagger} - g_i^{\dagger}) \cdot h_j \cdot v^{i-j} \bmod q_3$$

and

$$\left\|\mathbf{z}^{\dagger}\right\| \leq \max\{\alpha_6^*, t \cdot s \cdot q_0 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^2\} \leq \beta_{q_3}/2$$

On the other hand, if we define $\hat{\mathbf{z}}_{-0} = (\hat{z}_{-s}, \ldots, \hat{z}_{-1}, \hat{z}_0, \hat{z}_1, \ldots, \hat{z}_s)$ as

$$\begin{aligned}
\hat{z}_0 &\coloneqq 0 \\
\hat{z}_k &\coloneqq \sum_{i,j\in[t],i-j=k} \left(\hat{e}_j \cdot f_i^{\dagger} + q_0 \cdot r_i^{\dagger} - g_i^{\dagger}\right) \cdot h_j \quad \text{for } k \in \pm[s]
\end{aligned}$$

we have that

$$(\bar{\mathbf{v}}||1||\mathbf{v})^{\mathsf{T}} \cdot \hat{\mathbf{z}} = \sum_{i,j\in[t],i\neq j} (\hat{e}_j^{\dagger} \cdot f_i^{\dagger} + q_0 \cdot r_i^{\dagger} - g_i^{\dagger}) \cdot h_j \cdot v^{i-j} \bmod q_3,$$

and

$$\begin{aligned}
\|\hat{\mathbf{z}}\| &\leq t \cdot q_0/2 \cdot \alpha_2^* \cdot \alpha_3^* \cdot q_1 \cdot \gamma_{\mathcal{R}}^3 + q_0 \cdot q_1 \cdot \alpha_5 \cdot \gamma_{\mathcal{R}}^2 + q_0 \cdot \alpha_4 \cdot \gamma_{\mathcal{R}} \\
&\leq t \cdot q_0 \cdot q_1 \cdot (\alpha^*)^2 \cdot \gamma_{\mathcal{R}}^3 \\
&\leq \beta_{q_3}/2
\end{aligned}$$

Therefore

$$\left\langle (\bar{\mathbf{v}}||1||\mathbf{v}), \hat{\mathbf{z}} - \mathbf{z}^{\dagger} \right\rangle = 0 \bmod q_3 \qquad \text{and} \qquad \left\|\hat{\mathbf{z}} - \mathbf{z}^{\dagger}\right\| \leq \beta_{q_3}.$$

One (or both) of the following two cases must be true

(i) $\sum_{i\in[t]} h_i \cdot (\hat{e}_i \cdot f_i^\dagger + q_0 \cdot r_i^\dagger + g_i^\dagger) = 0$ with non-negligible probability.

(ii) $\sum_{i\in[t]} h_i \cdot (\hat{e}_i \cdot f_i^\dagger + q_0 \cdot r_i^\dagger + g_i^\dagger) \neq 0$ with non-negligible probability,

If Case (i) is true, we have

$$\sum_{i\in[t]} h_i \cdot (\hat{e}_i \cdot f_i^\dagger + q_0 \cdot r_i^\dagger + g_i^\dagger) = 0 \bmod q_1 \quad \text{and} \quad 0 < \left\|(\hat{e}_i \cdot f_i^\dagger + q_0 \cdot r_i^\dagger + g_i^\dagger)_{i\in[s]}\right\| \leq \beta_{q_1}$$

with non-negligible probability. This contradicts Assumption 7. If Case (ii) is true, we have

$$\left\langle (\bar{\mathbf{v}}||1||\mathbf{v}), \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\rangle = 0 \bmod q_3 \qquad \text{and} \qquad 0 < \left\|\hat{\mathbf{z}} - \mathbf{z}^\dagger\right\| \leq \beta_{q_3}$$

with non-negligible probability. This contradicts Assumption 9. Since none of the two cases could be true, we must have $(\mathbf{E} \cdot \mathbf{x}^\dagger) \circ (\mathbf{F} \cdot \mathbf{x}^\dagger) = \mathbf{G} \cdot \mathbf{x}^\dagger \bmod q_0$, as claimed. □

C.7.3 Efficiency

Theorem C.7.3. *Let* $n = \max\{|\mathbf{E}|, |\mathbf{F}|, |\mathbf{G}|, s+t\}$, $\eta, \alpha, \beta, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ *be a fixed polynomial in* λ, $(q_0, q_1, q_2, q_3) = (s, s^2, t \cdot s^4, (s+t)^{14}) \cdot \mathsf{poly}(\lambda)$, *and* $m = \log n \cdot \mathsf{poly}(\lambda)$. *Then* $\Pi^{\mathtt{bin\text{-}sat}}$ *has 1. common reference string size* $O_\lambda(n \cdot \log n)$, *2. proof size* $O_\lambda(\log^2 n)$, *3. prover time* $O_\lambda(n \cdot \log^3 n)$, *4. preprocessing time* $O_\lambda(n \cdot \log^2 n)$, *and 5. verifier time* $O_\lambda(\log^3 n)$ *after preprocessing.*

Proof. Note that $\log |\mathcal{R}_{q_3}| = \log q_3^{\varphi(\rho)} = O_\lambda(\log q_3) = O_\lambda(\log n)$, and an $\mathcal{R}_q$ operation takes at most $O_\lambda(\log^2 n)$ bit operations. Notice that $\mathbf{u}_\mathbf{E}, \mathbf{u}_\mathbf{F}, \mathbf{u}_\mathbf{G}, \mathbf{u}_\mathbf{z}$ can be computed in time $O_\lambda(n \cdot \log^3 n)$, exploiting fast multiplication algorithms for Toeplitz matrices (similarly to what described in Appendix C.5.3). All claims then follow by the same calculations as in Theorem 4.8.3. □

C.8 Argument for Succinct-R1CS

In this section, we describe a folding-based succinct argument for succinct-R1CS [BCG+19], which captures computations involving iterative executions of small circuits, with quasi-linear-time prover and polylogarithmic-time verifier without preprocessing. The high-level idea of the construction is identical to that in Section 4.9, except that here we will consider linear relations represented by not just a single, but multiple, foldable matrices. To avoid distraction by having too many variables, we only provide a sketch of the construction.

Recall that a succinct-R1CS instance is given by $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{y})$ and a witness $\mathbf{x}$ satisfies

$$(\mathbf{A} \cdot \mathbf{x}) \circ (\mathbf{B} \cdot \mathbf{x}) = (\mathbf{C} \cdot \mathbf{x}) \bmod q_0$$
$$\mathbf{D} \cdot \mathbf{x} = \mathbf{y} \bmod q_0$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ representing the "time constraints" are of the form

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & & & \\ & \mathbf{A}_0 & \mathbf{A}_1 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{A}_0 & \mathbf{A}_1 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} \mathbf{B}_0 & \mathbf{B}_1 & & & \\ & \mathbf{B}_0 & \mathbf{B}_1 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{B}_0 & \mathbf{B}_1 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 & & & \\ & \mathbf{C}_0 & \mathbf{C}_1 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{C}_0 & \mathbf{C}_1 \end{pmatrix},$$

and $(\mathbf{D}, \mathbf{y})$ represents the "boundary constraints". In the following, we outline a folding protocol for a variant of succinct-R1CS over $\mathcal{R}$ where $\mathbf{x}$ additionally satisfies a bounded-norm constraint $\|\mathbf{x}\| \leq \alpha$ and $\mathbf{D}$ (after removing the first and last block-columns) is foldable.

Let $s = w(n+2)$ denote the number of columns in $\mathbf{A}$ (and hence also in $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$). Similar to the strategy for proving R1CS, we let the prover commit to $\mathbf{a} = \mathbf{A} \cdot \mathbf{x}$, $\mathbf{b} = \mathbf{B} \cdot \mathbf{x}$, and $\mathbf{c} = \mathbf{C} \cdot \mathbf{x}$ as

- $\bar{c}_{\mathbf{h} \circ \mathbf{a}} = \bar{\mathbf{v}}^{\mathsf{T}} \cdot (\mathbf{h} \circ \mathbf{a}) \bmod q_3$,
- $c_{\mathbf{b}} = \mathbf{v}^{\mathsf{T}} \cdot \mathbf{b} \bmod q_3$, and
- $c_{\mathbf{c}} = \mathbf{v}^{\mathsf{T}} \cdot \mathbf{c} \bmod q_3$

respectively, where $\mathbf{h}$ is a foldable vector of norm $q_0 \ll \|\mathbf{h}\| \ll q_3$, and prove that

$$\begin{pmatrix} \mathbf{A} & -\mathbf{I} & & \\ \mathbf{B} & & -\mathbf{I} & \\ \mathbf{C} & & & -\mathbf{I} \\ \mathbf{D} & & & \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{y} \end{pmatrix} \bmod q_0, \tag{C.1}$$

$$\begin{pmatrix} \mathbf{0} & (\bar{\mathbf{v}} \circ \mathbf{h})^{\mathsf{T}} & & \\ \mathbf{0} & & \mathbf{v}^{\mathsf{T}} & \\ \mathbf{0} & & & \mathbf{v}^{\mathsf{T}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \bar{c}_{\mathbf{h} \circ \mathbf{a}} \\ c_{\mathbf{b}} \\ c_{\mathbf{c}} \end{pmatrix} \bmod q_3, \tag{C.2}$$

and $\|(\mathbf{x}, \mathbf{a}, \mathbf{b}, \mathbf{c})\| \approx 0$. Observe that Equation (C.1) is equivalent to $\mathbf{A} \cdot \mathbf{x} = \mathbf{a}$, $\mathbf{B} \cdot \mathbf{x} = \mathbf{b}$, $\mathbf{C} \cdot \mathbf{x} = \mathbf{c}$, and $\mathbf{D} \cdot \mathbf{x} = \mathbf{y}$ all modulo q_0, whereas Equation (C.2) ensures that the commitments $\bar{c}_{\mathbf{h} \circ \mathbf{a}}$, $c_{\mathbf{b}}$, and $c_{\mathbf{c}}$ are well-formed. Then, we let the prover prove that $\mathbf{a} \circ \mathbf{b} = \mathbf{c}$ by proving the existence of

$$\mathbf{z} = \left(\sum_{0 \leq i,j, \leq s : j - i = k} h_i a_i b_j - h_i c_j \right)_{-s \leq k \leq s}$$

which satisfies

$$(\bar{\mathbf{v}} \mid \mathbf{v})^{\mathrm{T}} \cdot \mathbf{z} = \bar{c}_{\mathbf{h} \circ \mathbf{a}} \cdot c_{\mathbf{b}} - \bar{c}_{\mathbf{h}} \cdot c_{\mathbf{c}} \bmod q_3 \qquad \text{and} \qquad \|\mathbf{z}\| \approx 0. \tag{C.3}$$

Since Equations (C.2) and (C.3) are represented by foldable matrices, an adaption of the folding protocol in Section 4.7 applies. For Equation (C.1), we need to handle one technical issue: The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are not in the block-bidiagonal form which is supported by the folding protocol in Section 4.7. Taking $\mathbf{A}$ as an example, we observe that we have one $\mathbf{A}_0$ block extra at the top left, and one $\mathbf{A}_1$ block extra at the bottom right. To deal with this issue, we let the prover reveal the first and last blocks of $\mathbf{x}$, so that the verifier can subtract the contributions of these blocks from $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$. Letting $\mathbf{A}'$, $\mathbf{B}'$, $\mathbf{C}'$, and $\mathbf{D}'$ be derived from their counterparts with the first and last block-columns removed, we obtain a relation of the form

$$\begin{pmatrix} \mathbf{A}' & -\mathbf{I} & & \\ \mathbf{B}' & & -\mathbf{I} & \\ \mathbf{C}' & & & -\mathbf{I} \\ \mathbf{D}' & & & \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{y}_a \\ \mathbf{y}_b \\ \mathbf{y}_c \\ \mathbf{y}_d \end{pmatrix} \bmod q_0,$$

where $\mathbf{A}' = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_0 \end{bmatrix}_{\searrow n}$, $\mathbf{B}' = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_0 \end{bmatrix}_{\searrow n}$, $\mathbf{C}' = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_0 \end{bmatrix}_{\searrow n}$, $\mathbf{D}'$, $\mathbf{y}_a$, $\mathbf{y}_b$, $\mathbf{y}_c$, and $\mathbf{y}_d$ are foldable. We can therefore adapt the folding protocol in Section 4.7 to prove the statement.