Immunizing Hash-Based Signatures from Backdoored Hash Functions

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Lukas Brandstetter
Registration Number 01307003

to the Faculty of Informatics

at the TU Wien

Advisor: Thomas Grechenig
Assistance: Clemens Hlauschek

at the TU Darmstadt

Advisor: Marc Fischlin

Vienna, 2nd July, 2021

Signature Author

Signature Advisor

TECHNISCHE
UNIVERSITÄT
DARMSTADT

TU WIEN Informatics

Immunizing Hash-Based Signatures from Backdoored Hash Functions

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

IT-Sicherheit

by

Lukas Brandstetter
Registration Number 2495888

to the Department of Computer Science

at the TU Darmstadt

Advisor: Marc Fischlin

at the TU Wien

Advisor: Thomas Grechenig
Assistance: Clemens Hlauschek

Vienna, 2^{nd} July, 2021

Signature Author

Signature Advisor

Double Degree IT Security ▪ Technische Universität Darmstadt & Technische Universität Wien
www.tu-darmstadt.de ▪ www.tuwien.at

Immunizing Hash-Based Signatures from Backdoored Hash Functions

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Lukas Brandstetter
Registration Number 01307003

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Advisor: Marc Fischlin

Wien, 2nd July, 2021

Erklärung zur Verfassung der Arbeit

Lukas Brandstetter

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 2. Juli 2021

Lukas Brandstetter

Abstract

The Snowden revelations raised awareness of a possible attack vector against cryptographic schemes: the embedding of a backdoor in design and specification. This may have already been done — it seems likely that the Pseudorandom Generator (PRG) DUAL_EC_DRBG was designed and standardized with a deliberate backdoor [Che+14; BLN15; Che+16]. We need to assume that this will not remain the only instance of a malicious party pushing a backdoored cryptographic standard.

While DUAL_EC_DRBG is a PRG, nothing prevents one from also embedding a backdoor in hash functions or signatures in a similar way. Signatures are used in protocols such as Transport Layer Security (TLS) [Res18] underlying security of the internet of today or to verify software updates before installation. Including a backdoor in a signature standard may allow adversaries to intercept and modify traffic secured by TLS or attack targets via malicious software updates. Doing so thus provides a valuable objective and promising outlook for malicious parties.

This thesis provides a formal treatment of techniques to immunize signatures from included backdoors. We focus on hash-based signatures schemes starting from few-time signature and building up to many-time signatures for long messages. Two results we provide are essential for the immunization of hash-based signatures. First, it is implausible to include a backdoor in the pseudorandomness notion of an efficient PRG, that would allow an adversary with the backdoor key to distinguish the PRG from random. Second, it is implausible to include a backdoor in an efficient hash functions without revealing the backdoor by using it. Such a backdoor would allow an adversary with a backdoor key to produce collisions or second-preimages for the hash function.

Keywords: Signatures, Hash-Based Signatures, Backdoors, Malicious Hashing, PRG

Kurzfassung

Die Snowden Enthüllungen zeigten einen neuen potenziellen Angriffsvektor auf kryptographische Verfahren: Hintertüren in Design und Spezifikation. Das ist möglicherweise schon durchgeführt worden – es ist wahrscheinlich, dass der PRG DUAL_EC_DRBG mit einer bewussten Hintertür spezifiziert wurde [Che+14; BLN15; Che+16]. Wir müssen annehmen, dass dies nicht der einzige Fall bleiben wird, in dem eine maliziöse Partei einen kryptographischen Standard forciert.

Auch wenn DUAL_EC_DRBG ein PRG ist, gibt es nichts das prinzipiell verhindert auch ähnliche Hintertüren in Hash-Funktionen und Signaturen einzubauen. Signaturen werden in Protokollen wie TLS [Res18] verwendet, die ein Fundament für die Sicherheit des heutigen Internets bilden, sowie zur Verifikation von Software Updates vor der Installation. Hintertüren in Signatur-Standards könnten maliziösen Parteien erlauben, Datenverkehr zu überwachen und sogar zu modifizieren oder bösartige Software Updates ausliefern. Solche standardisierten Hintertüren sind daher vielversprechende Ziele für maliziöse Parteien.

Diese Arbeit beschäftigt sich formal mit Techniken um Signaturverfahren gegen Hintertüren zu immunisieren. Dabei liegt der Fokus auf Hash-Basierten Signaturverfahren. Angefangen bei Wenige-Male Signaturen und aufbauend bis zu Viele-Male Signaturen für lange Nachrichten. Zwei Resultate dieser Arbeit sind essentiell für die Immunisierung von Hash-Basierten Signaturverfahren. Erstens scheint es unwahrscheinlich, erfolgreich Hintertüren für die Pseudozufälligkeit eines effizienten PRG zu konstruieren, die einem Angreifer erlauben, die Ausgabe des PRG von einem zufälligen Bitstring zu unterscheiden. Zweitens scheint es unwahrscheinlich, erfolgreich Hintertüren für effiziente Hash-Funktionen zu konstruieren, bei dennen die Hintertür nicht bei Verwendung auffällt. Eine Hintertür in diesem Fall würde einem Angreifer erlauben Kollisionen und Zweite Urbilder zu generieren.

Keywords: Signatures, Hash-Based Signatures, Backdoors, Malicious Hashing, PRG

Contents

CHAPTER 1

Introduction

Recent works analysed how backdoors in the design of hash functions are a threat to the security notions of those hash functions and how to defend against this threat [FJM18; BFM18; Dod+20]. HMAC [KBC97] and HKDF [KE10], both based on the security of hash functions, were immunized via an implausibility of backdoors result for pseudorandomness properties of the hash function used [FJM18]. Examples of backdoors included in the design of hash functions were also discussed in the same paper.

Hash functions are important building blocks for many higher level cryptographic constructions like message authentication codes, key derivation functions and signature schemes. These constructions often rely on security notions of the hash function for their own security reductions. Instantiating any of these with a backdoored hash function might compromise the security of the construction as a whole. A natural question is thus whether the security of other primitives is threatened by backdoors included in the design and what we can do to immunize against this threat.

Hash-based signature schemes are promising post-quantum signature candidates because their (often minimal) security assumptions are conjectured to hold even against adversaries with access to large-scale quantum computers. For this reason, we set out to analyse the possibility of backdoors in hash-based signatures and immunization techniques against that possibility. We claim that it is possible to immunize hash-based signatures against backdoors by sacrificing a bit of performance compared to optimized practical proposals.

1.1 Contributions

Backdoored Hash Functions. Fischlin, Janson and Mazaheri [FJM18] describe backdoored compression functions that allow an adversary with backdoor key to violate (second-)preimage resistance and collision resistance of the compression function. In those functions, the adversary must include the backdoor key in every preimage or collision produced with the backdoor. We ask whether revealing the backdoor key is necessary for every backdoor of this kind and provide arguments that this is the case. Our argument is that hash functions with reusable backdoors (i.e., evidence of an attack does not impede security against adversaries without backdoor key) imply hash-based signature schemes with short signatures and verification times. Signatures with these properties are unknown and seem implausible and we conclude that hash functions with reusable backdoors are implausible too. We show this implication for second-preimage resistance and collision resistance. In order to prove it we also provide an approach to modelling reusable backdoors for hash functions that is in principle adaptable to other cryptographic security notions. Second, we improve a recent result that combiners for backdoored random oracles retain preimage resistance and pseudorandomness [BFM18] by showing that the concatenation combiner additionally retains second-preimage resistance. We also reprove, in a slightly different model and with a fix to the construction, that it is implausible to include a backdoor in a PRG [Dod+15]. Here the argument is that a backdoored PRG can be used to construct efficient public key encryption.

Immunizing Hash-Based Signatures Equipped with the aforementioned immunizations of primitives we instantiate signature constructions and build up to hash-based signatures of long messages. Starting from a PRG, we use preimage resistance of expanding PRG constructions to instantiate the few-time signature scheme HORS++ [PWX04]. If we leave the standard model and assume backdoored random oracles we can also instantiate WOTS^+ [Hül13], an efficient and practical one-time signature scheme used for example in XMSS-T [HRS16] and $\mathsf{SPHINCS}^+$ [Ber+19]. Moving from few-time signatures to many-time signatures we can instantiate XMSS with the assumption that adversaries need reusable backdoors and an immunized HMAC [FJM18] used as a Pseudorandom Function (PRF). If we do not assume that adversaries need reusable backdoors, XMSS is still possible to instantiate with combiners of backdoored random oracles. The above schemes can only sign short messages. We describe how to extend short message signatures via standard techniques, again instantiated with either standard hash functions under reusable backdoors or combiners of backdoored random oracles.

1.2 Notation

The set of all n-bit strings is denoted by $\{0,1\}^n$ and the set of bit strings of arbitrary lengths by $\{0,1\}^*$. $\perp$ is a special symbol denoting an error or missing value. The length of a bit string s is denoted by $|s|$ and the concatenation of strings s_1 and s_2 by $s_1\|s_2$. The set of all functions from $f : \{0,1\}^{il} \to \{0,1\}^{ol}$ is denoted by $\mathsf{Fun}[il, ol]$. We name the length of an input il, the length of an output ol and the length of the key kl if possible and unambiguous within a section.

When describing algorithms $\leftarrow\$$ is used to assign return values of possibly probabilistic subroutines or algorithms and $=$ is used if the subroutine or algorithm must be deterministic. Some algorithms have access to an oracle that they can query denoted by $\mathcal{A}^{\mathsf{O}}$ where $\mathcal{A}$ is the algorithm having access to oracle O.

$\Pr[E]$ denotes the probability that the event E happens where the event can be the return value of a probabilistic algorithm. If the algorithm $A(x)$ returns a single bit (interpreted as a boolean) we often write $\Pr[A(x)]$ as the probability that $A(x)$ returns 1 (interpreted as true).

A negligible function $\varepsilon : \mathbb{N} \to \mathbb{R}$ if for any polynomial $\mathsf{poly} : \mathbb{N} \to \mathbb{R}^+$ there is a $N \in \mathbb{N}$ such that for all $n \geq N$ $\varepsilon(n) \leq \frac{1}{\mathsf{poly}(n)}$. We call a probability negligible if it is described by a negligible function. A noticeable function $\delta : \mathbb{N} \to \mathbb{R}$ if there exists a polynomial $\mathsf{poly} : \mathbb{N} \to \mathbb{R}^+$ there is a $N \in \mathbb{N}$ such that for all $n \geq N$ $\varepsilon(n) \geq \frac{1}{\mathsf{poly}(n)}$. We call a probability noticeable if it is described by a noticeable function. We call a probability overwhelming if there exists a negligible function ε and the probability can be described by $1 - \varepsilon(n)$. Note that a non-negligible function is not necessarily noticeable and vice versa. The function

$$f(n) = \begin{cases} 2^{-n} & \text{for even } n \\ n^{-2} & \text{for odd } n \end{cases}$$

is neither negligible nor noticeable because for even n it not noticeable and for odd n it is not negligible. A non-negligible probability thus only guarantees that the probability is greater than some inverse polynomial for infinitely many n.

1.3 Structure of this Thesis

Chapter 2 gives an overview over related work and Chapter 3 establishes security notions needed throughout the rest of this thesis and reviews the hash-based signatures that we either instantiate or are at least inspired by in later sections.

The next part of this thesis, Chapter 4 contains the implausibility results regarding hash functions with reusable backdoors. First intuition for our way of modelling reusable backdoors is established. The approach is then applied to the second-preimage resistance and collision resistance notions of security. Afterwards we discuss alternative possible formalizations options and give reasoning of why we decided against these alternative approaches.

Chapter 5 then starts immunizing the first step to hash-based signatures, namely few-time signatures. Concretely, the notion of PRG is shown to have implausibility of backdoors and such a PRG is then used to instantiate the $\mathsf{HORS}{+}{+}$ few-time signature scheme. Alternatively, we show here that combiners of backdoored random oracles also retain second-preimage resistance and can thus be used to instantiate WOTS^+.

Chapter 6 and Chapter 7 continues by immunizing Merkle signature schemes for long messages by instantiating known constructions with either combiners of backdoored random oracles again or by using standard hash functions and relying on implausibility results of Chapter 4.

Chapter 8 shows examples of how backdoors can indeed impede security of signature schemes and serves as validation of the importance of treatment of backdoors in hash-based signatures.

CHAPTER 2

Related Work

Backdoored Secret-Key Primitives Examples of backdoored symmetric primitives, i.e., with backdoors included in the design, exist. A well known case is the PRG DUAL_EC_DRBG, possibly designed and standardized with a deliberate backdoor [Che+14; BLN15; Che+16]. Besides pseudorandom generators, symmetric ciphers designs that include different variations of a backdoor were presented, allowing key recovery via linear cryptanalysis methods [RP97; PA21] or via differential cryptanalysis methods [PW20]. Malicious variants of SHA-1 [Alb+14], SHA-3 [Mor15] and Streebog [AY15] were proposed. Those variants deliberately choose round constants of the respective hash functions to allow the designer to generate collisions. For SHA-1 and Streebog the round constants are chosen such that the designer knows a single unique collision. SHA-1 and Streebog are iterated hash functions and a single collision is sufficient to generate many more collisions by extending both inputs with the same blocks. A single collision however is not sufficient to constitute a backdoored hash function in the model we introduce later. The malicious round constants for SHA-3 allow more efficient attacks by abusing symmetry properties. The attacks do not require a special backdoor and still have time complexity super polynomial in the security parameter. For BLAKE a similar strategy was proposed [Aum11], modifying finalization functions instead of round constants to be able to generate collisions. Again the finalization function is adapted for a specific collision that is known at design time and is not sufficient to constitute a backdoored hash function in our model. Even if malicious hash functions were designed before none of the backdoors can be used multiple times without revealing enough information to give other parties essentially the same power. For this reason the malicious hash functions do not violate our results.

Backdoors Implementations Young and Yung initiated the study of backdoors in cryptographic implementations under the term kleptography [YY96; YY97]. The study of Algorithm Substitution Attack (ASA) [BPR14] is a recent iteration of the idea of kleptography. Feasability of subverting implementations via an ASA as well as preventing these types of attacks was first studied for symmetric primitives [BPR14; BJK15] and later extended to different primitives, e.g., message authentication codes [AP19] and public key encryption [BH15]. The heartbleed [Cveb] and Debian SSH vulnerability [Cvea] are notable examples of (probably) unintentional ASAs encountered in practice as they may allow an adversary to recover secret key material from otherwise secure implementations. Distinct from the aforementioned works, this thesis assumes backdoors in the design of algorithms rather then in concrete implementations of a well designed algorithm. Both approaches are necessary in order to achieve the goal of deploying backdoor free cryptography in practice. Implementations immune against ASAs cannot achieve this goal because the design might be backdoored and a design immune against backdoors might still be implemented with a backdoor.

Implausibility of Backdoored Standard Model Constructions. Fischlin, Janson, and Mazaheri immunize HMAC and HKDF with implausibility of backdoors [FJM18]. A similar result has been shown regarding a PRG [Dod+15]. The proof strategy is the same in both cases: The backdoored primitive implies public key encryption. The primitives used are a weak form of a PRF [FJM18] and a stateful PRG [Dod+15]. Both are known to be implied by one-way functions, i.e., if one-way functions exist, PRFs and PRGs can be built from them in a black-box way. This implication does not hold for public key encryption, separating one-way functions, PRGs and PRFs into one class and public key encryption into another. Thus, it seems implausible to build a backdoored PRF or PRG with a non obvious backdoor. The resulting schemes seem to be efficient compared to current standards in public key encryption (depending on the exact advantage and runtime of the adversary with backdoor key). In this theses, we build on these results and use them to construct different higher level primitives, namely hash-based signatures.

Circumventing Backdoored Random Oracles. A different way to reach primitives that are implausible to backdoor is via idealized models instead of the standard model. Backdoored random oracles (BRO) are random oracles that provide the adversary access to a backdoor oracle that computes arbitrary functions on the function table of the random oracle and returns the result. Results from communication complexity can be used to prove that combining two (independently) backdoored random oracles via concatenation or composition retains at least preimage resistance and pseudorandomness [BFM18]. Later work even showed indifferentiability of combiners from a random oracle as long as the adversary is restricted in the number of times it can switch between the backdoor oracles [Dod+20]. In this thesis, we use the results for preimage resistance and pseudorandomness and prove that second-preimage resistance is also retained by combining random oracles via concatenation.

Hash-Based Signatures. XMSS renewed interest in hash-based signatures as candidates for post-quantum cryptography. Hash-based signatures provide comparable performance to currently used signatures [Bos+21] and are implementable even on microcontrollers [Cam+20] and embedded systems [Wan+19]. XMSS uses a variant of WOTS [Mer90a; Buc+11b] one-time signatures as leaves of a Merkle tree. The tree structure is used to combine the individual WOTS instances into a single verification key. XMSS is standardized by the IETF [Hül+18]. Using one-time signatures means that special care must be taken that these one-time signatures are not reused. Reuse would have catastrophic consequences for the overall construction, in most cases allowing an adversary to forge signatures for arbitrary message. XMSS solves this by being a stateful construction, where the state includes information about already used one-time signatures. Additionally, only a predetermined number of signatures can be generated under a single XMSS signing key. XMSS$^{\mathsf{MT}}$ [HRB13] uses multiple layers of trees. Each leaf signatures of an upper level signs the root of the lower level trees. This allows a theoretically infinite number of signatures verifiable with the same verification key. Having to manage state is the main issue of early practical hash-based signatures. An issue that SPHINCS solved by randomizing the selection of the leaf and using few-time signatures instead of one-time signatures as those leaves. SPHINCS^{+} [Ber+19], a variant of SPHINCS, is a third round alternative candidate for the NIST post quantum cryptography competition [NIS].

CHAPTER 3

Preliminaries

Hash-based signatures emerged in recent years as promising candidates for practical signatures schemes [BDH11; HRS16; Ber+15; Ber+19] for two important reasons. They allow signature schemes to be secure with the minimal assumption of preimage resistant functions (or one-way functions) and they are conjectured to be secure even in the face of adversaries with access to a quantum computer. We review definitions of backdoored hash functions [FJM18], few-time signature schemes and Merkle signature schemes. The concrete Merkle signature schemes we will draw inspiration from are XMSS [BDH11], XMSS-T [HRS16], SPHINCS [Ber+15] and SPHINCS$^+$ [Ber+19]. All of them use few-time (or one-time) signature schemes as leaf nodes in a Merkle tree that allows them to verify the signatures under a single verification key.

3.1 Hash Functions

Cryptographic hash functions lie at the core of our analysis. We define keyed hash functions and their most common security notions, preimage resistance, second-preimage resistance and collision resistance with backdoors in their game based definition [FJM18; Maz20]. There is a weaker backdoor model in which the hash function generator can specify the randomness used in the hash key generation but we only consider the stronger version here. We call the hash functions keyed or unkeyed explicitly to distinguish between different definitions of the security notion. Some results in this work are stated in terms of fixed input length compression functions instead of full hash functions. Many practical variable input length hash functions are constructed from compression functions iterated via domain extensions like the Merkle-Darmgård construction [Dam90; Mer90b]. Iterating compression functions is not the only way to build variable input length hash functions. A recent example of a different methodology is the sponge construction used by SHA-3 [Dwo15].

```
$\mathrm{PRE}^{bd}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)$
$(\mathsf{H}, \mathsf{bk}) \leftarrow\$ \mathsf{BDHGen}(1^\lambda)$
**if** $bd = 0$ **then** $\mathsf{bk} = \bot$
$\mathsf{k} \leftarrow\$ \{0,1\}^{kl}$
$x \leftarrow\$ \{0,1\}^{il}$
$y = \mathsf{H}_\mathsf{k}(x)$
$x' \leftarrow\$ \mathcal{A}(\mathsf{bk}, \mathsf{H}, \mathsf{k}, y)$
**return** $\mathsf{H}_\mathsf{k}(x') = \mathsf{H}_\mathsf{k}(x)$
```

```
$\mathrm{SPR}^{bd}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)$
$(\mathsf{H}, \mathsf{bk}) \leftarrow\$ \mathsf{BDHGen}(1^\lambda)$
**if** $bd = 0$ **then** $\mathsf{bk} = \bot$
$\mathsf{k} \leftarrow\$ \{0,1\}^{kl}$
$x \leftarrow\$ \{0,1\}^{il}$
$x' \leftarrow\$ \mathcal{A}(\mathsf{bk}, \mathsf{H}, \mathsf{k}, x)$
**return** $\mathsf{H}_\mathsf{k}(x') = \mathsf{H}_\mathsf{k}(x)$
        $\wedge\, x \neq x'$
```

```
$\mathrm{CR}^{bd}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)$
$(\mathsf{H}, \mathsf{bk}) \leftarrow\$ \mathsf{BDHGen}(1^\lambda)$
**if** $bd = 0$ **then** $\mathsf{bk} = \bot$
$\mathsf{k} \leftarrow\$ \{0,1\}^{kl}$
$(x, x') \leftarrow\$ \mathcal{A}(\mathsf{bk}, \mathsf{H}, \mathsf{k})$
**return** $\mathsf{H}_\mathsf{k}(x') = \mathsf{H}_\mathsf{k}(x)$
        $\wedge\, x \neq x'$
```

Figure 3.1: Security games for keyed and unkeyed hash functions

We use definitions that include backdoor capabilities taken from [FJM18]. The bit bd decides whether the adversary is given a backdoor key or not. Note that setting $bd = 0$ in any of the following definitions amounts to the standard definition of preimage resistance, second-preimage resistance or collision resistance respectively. It is thus possible to model standard hash functions without backdoors with the same definitions.

Definition 1 (Hash Function Generator)**.** A PPT algorithm BDHGen is called a hash function generator if on input a security parameter 1^λ, it outputs a family of PPT keyed hash functions $\mathsf{H} = \{\mathsf{H}_\mathsf{k} : \{0,1\}^{il} \to \{0,1\}^{ol} \mid \mathsf{k} \leftarrow\$ \{0,1\}^{kl}\}$ and a backdoor key bk, for length parameters $il, ol, kl \in \mathbb{N}$ with $il \geq ol$ and depending on the security parameter 1^λ. H is called a keyed hash function. H is called an unkeyed hash function if it ignores the key, i.e., the family is a single function.

Definition 2 (Preimage Resistance)**.** A hash function H provides preimage resistance (with backdoor for $bd = 1$) if for every PPT adversary $\mathcal{A}$ its advantage in the PRE game in Fig. 3.1 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{PRE},bd}_{\mathsf{BDHGen},\mathcal{A}}(\lambda) = \Pr\left[\mathrm{PRE}^{bd}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)\right]$$

The probability is over the choice of k and x and the random coins of $\mathcal{A}$.

Definition 3 (Second-Preimage Resistance)**.** A hash function H provides second-preimage resistance (with backdoor for $bd = 1$) if for every PPT adversary $\mathcal{A}$ its advantage in the SPR game in Fig. 3.1 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{SPR},bd}_{\mathsf{BDHGen},\mathcal{A}}(\lambda) = \Pr\left[\mathrm{SPR}^{bd}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)\right]$$

The probability is over the choice of k and x and the random coins of $\mathcal{A}$.

Definition 4 (Collision Resistance)**.** A hash function H provides collision resistance (with backdoor for $bd = 1$) if for every PPT adversary $\mathcal{A}$ its advantage in the CR game in Fig. 3.1 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{CR},bd}_{\mathsf{BDHGen},\mathcal{A}}(\lambda) = \Pr\left[\mathrm{CR}^{bd}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)\right]$$

The probability is over the choice of k and the random coins of $\mathcal{A}$.

Remark. At least the notion of collision resistance needs keyed hash functions. If an unkeyed hash function compresses its input, i.e. $il > ol$, it is impossible to achieve collision resistance against non-uniform adversaries without some type of key. Via pigeonhole principle, we know that at least one collision exists. An adversary against an unkeyed hash function can simply include a collision in its advise and return it.

Chapter 5 describes a construction of an unkeyed hash function with preimage resistance, often called a one-way function. The security game stays the same for unkeyed hash functions. The hash function H simply ignores the key.

Definition 5 (S-Backdoored Hash Function)**.** Let BDHGen be a PPT hash function generator and let $S \in \{\mathrm{PRE}, \mathrm{SPR}, \mathrm{CR}\}$ denote a security notion for hash functions. We call BDHGen a S-backdoored hash function generator and its output H a S-backdoored hash function, if there is a PPT adversary $\mathcal{B}$ such that the advantage $\mathsf{Adv}^{S,1}_{\mathsf{BDHGen},\mathcal{B}}(\lambda)$ is non-negligible and for every PPT adversary $\mathcal{A}$ the advantage $\mathsf{Adv}^{S,0}_{\mathsf{BDHGen},\mathcal{A}}(\lambda)$ is negligible.

3.2 Signature Schemes

We first define the general notions of signature schemes before reviewing hash-based signature schemes. A signature scheme is defined by three algorithms: KGen generates signature and verification key, Sign generates a signature for a given message and Vf verifies whether a given signature is valid for a given message.

Definition 6 (Signature Scheme)**.** A triple of PPT algorithms $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$ is called a signature scheme if:

- $\mathsf{KGen}(1^\lambda)$: On input the security parameter 1^λ, KGen generates a verification key vk and corresponding signature key sk.
- $\mathsf{Sign}(\mathsf{sk}, m)$: On input the signature key sk and a message m, Sign returns a signatures σ such that $\mathsf{Vf}(\mathsf{vk}, m, \sigma) = 1$.
- $\mathsf{Vf}(\mathsf{vk}, m, \sigma)$: On input the verification key vk, a message m and a signature σ, Vf verifies whether σ is a valid signature for m.

```
$\text{EUF-CMA}_{\mathcal{A}}(1^\lambda)$
    $(\mathsf{vk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$
    $(m^*, \sigma^*) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{vk})$
    return $\mathsf{Vf}(\mathsf{vk}, m^*, \sigma^*) = 1$
              $\wedge\, m^* \notin Q$

Oracle $\mathsf{Sign}(\mathsf{sk}, m)$
    $Q = Q \cup \{m\}$
    $\sigma \leftarrow\!\!\$\ \mathsf{Sign}(\mathsf{sk}, m)$
    return $\sigma$
```

Figure 3.2: Security games for existantial unforgeability of signature schemes

The signature scheme is correct if the following equation holds for all vk, sk generated by KGen and all valid messages m.

$$\mathsf{Vf}(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = 1$$

The signature scheme is approximately correct if the probability $\Pr[\mathsf{Vf}(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m)]$ is overwhelming for all vk, sk generated by KGen and all valid messages m.

Security of hash-based signatures is defined via the inability of an adversary to provide an existential forgery.

Definition 7 (Existential Unforgeability)**.** A signature scheme provides existential unforgeability under choosen message attack (EUF-CMA) if for every PPT adversary $\mathcal{A}$ its advantage in the EUF-CMA game in Fig. 3.2 is negligible in the security parameter λ.

$$\mathsf{Adv}_{\mathcal{A}}^{\text{EUF-CMA}}(\lambda) = \Pr\left[\text{EUF-CMA}_{\mathcal{A}}(1^\lambda)\right]$$

The probability is over the choices of $(\mathsf{vk}, \mathsf{sk})$ and the random coins of $\mathcal{A}$ and $\mathcal{B}$.

The number of queries the adversary can use the Sign oracle for is restricted for few-time signatures. The EUF-n-CMA game for n-time signatures restricts $\mathcal{A}$ to do n queries at most. The hash-based many-time signature schemes we build and use in this work also have a restriction on the number of times the signature scheme can be used, although that number is much higher and can be increased via parameter choices of the signature scheme.

3.3 Hash-based Few-Time Signatures

We give an overview over the relevant constructions for few-time signatures (with one-time signatures being a special case of the former) with emphasis of the concrete assumptions they need in their proofs. We emphasise the assumptions because we will not be able to reach all of them with implausibility of backdoors in the standard model.

Variants of the Winternitz One-Time Signature scheme WOTS [Mer90a; DSS05] are used in the XMSS and XMSS-T constructions. WOTS variants generally base their security on a combination of pseudorandomness, preimage resistance and second-preimage resistance or even collision resistance of the underlying function. There is WOTS-PRF [Buc+11b] which is based on a PRF but its security reduction seems to be flawed as pointed out by Lafrance and Menezes [LM17] and acknowledged by a note on the ePrint version of the WOTS-PRF paper [Buc+11a]. Security of the most recent variant WOTS^+ (used e.g., in XMSS and SPHINCS) needs preimage resistance, second-preimage resistance and pseudorandomness.

Another option are HORS [RR02] and HORS++ [PWX04] (independently described as a leakage resilient signature scheme [KV09]) that are used in SPHINCS and $\mathsf{SPHINCS}^+$ as leaf signatures. Both use a set of secrets as a signature key. A preimage-resistant function is applied to every element in the set to get the verification key. Thus all of them need preimage resistance in their security reductions. The difference is in how the subset of secrets that corresponds to a message and is revealed as the signature of the message is chosen. HORS uses a collision-resistant hash function (or target-collision-resistant in the non-adaptive case) to choose the subset. HORS++ uses cover-free families instead of a collision-resistant hash function to select the subset used for signing. The same construction is described by Katz and Vaikuntanathan as a leakage-resilient one-time signature [KV09]. Cover-free families exist unconditionally and thus HORS++ only needs preimage resistance to prove security. However HORS++ is not as efficient as HORS.

3.4 Hash-based Many-Time Signatures

Merkle trees [Mer88] are a common way to combine many few-time signature schemes under a single verification key [BDH11; HRS16; Ber+15; Ber+19]. We call schemes that are composed in this way Merkle signature schemes. In those, leaves consist of few-time signature schemes that are used to sign messages. These leaves are hashed together to form the next level of the Merkle tree. Intermediate nodes are hashed together up to the root that serves as a single verification key. Signatures consist of the signature of the leaf level few-time signature scheme and a so called authentication path, the sibling hashes needed to recompute the path from the leaf to the root. Merkle signature schemes can be stateful (e.g., XMSS and XMSS-T) or stateless (e.g., SPHINCS and $\mathsf{SPHINCS}^+$). Stateful schemes keep information about already used leaf signatures as state across signatures to prevent reusing the few-time signature schemes too many times. Stateless schemes randomize the selection of leaf nodes and increase the number of times the leaf signatures can be used to probabilistically prevent too much reuse. Regardless of state, Merkle signature schemes in the literature rely on either collision resistance or second-preimage resistance to connect nodes in the tree.

The hash-based many-time signature schemes used in the following are stateful (or key-evolving signature schemes). They hold the index of the last used leaf signature as state to prevent reuse of the same leaf signature. The state can be modelled by letting the signature algorithm Sign return an updated signature key sk that is used in the next round.

CHAPTER 4

Implausibility of Hash Functions with Reusable Backdoors

Fischlin, Janson and Mazaheri [FJM18] show constructions of backdoored compression functions that are preimage resistance, second-preimage resistance and collision resistance without knowledge of the backdoor key. Knowledge of the backdoor key allows an efficient algorithm to compute preimages, second-preimages and collisions. The design uses a secure compression function and modifies this function such that it allows the adversary control over the output on input of the backdoor key. This allows an adversary to violate all three security notions for the same compression function. An important caveat with the design is that the backdoor is revealed on first use. Every preimage, second-preimage or collision is either valid for the original compression function or includes the backdoor key. Thus by observing the backdoor in use, one learns the backdoor key. In the following we show that including a reusable backdoor in an efficient compression function is implausible without revealing some information about the backdoor to observing parties. The results hold for second-preimage resistance and collision resistance security notions of the compression function. In light of this result, it seems to be a necessary property of efficient compression function backdoors to reveal the backdoor on use.

4.1 Modelling Reusable Backdoors

We start from backdoored hash functions in Definition 5. The backdoored hash function is given by a generator outputting the description of a hash function (family) and a backdoor key. We say the hash function is backdoored if there is an adversary $\mathcal{B}$ that can violate the security notions of the hash function given the backdoor key with non-negligible advantage. We model that the hash function retains its security notions even after an adversary $\mathcal{A}$ (without backdoor key) sees evidence of the backdoor. The evidence is modelled by giving the adversary $\mathcal{A}$ access to an oracle that returns transcripts between

```
$\mathrm{SPR}^{\mathcal{B}}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)$

$(\mathsf{H}, \mathsf{bk}) \leftarrow\$ \mathsf{BDHGen}(1^\lambda)$
$\mathsf{k} \leftarrow\$ \{0,1\}^{kl}$
$x \leftarrow\$ \{0,1\}^{il}$
$x' \leftarrow\$ \mathcal{A}^{\mathrm{SPR\text{-}BD}}(\mathsf{H}, \mathsf{k}, x)$
return $\mathsf{H}_\mathsf{k}(x') = \mathsf{H}_\mathsf{k}(x)$
        $\wedge\, x \neq x'$
```

```
Oracle SPR-BD()

$\mathsf{k} \leftarrow\$ \{0,1\}^{kl}$
$x \leftarrow\$ \{0,1\}^{il}$
$x' \leftarrow\$ \mathcal{B}(\mathsf{bk}, \mathsf{H}, \mathsf{k}, x)$
return $(\mathsf{k}, x, x')$
```

```
$\mathrm{CR}^{\mathcal{B}}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)$

$(\mathsf{H}, \mathsf{bk}) \leftarrow\$ \mathsf{BDHGen}(1^\lambda)$
$\mathsf{k} \leftarrow\$ \{0,1\}^{kl}$
$(x, x') \leftarrow\$ \mathcal{A}^{\mathrm{CR\text{-}BD}}(\mathsf{H}, \mathsf{k}, x)$
return $\mathsf{H}_\mathsf{k}(x') = \mathsf{H}_\mathsf{k}(x)$
        $\wedge\, x \neq x'$
```

```
Oracle CR-BD()

$\mathsf{k} \leftarrow\$ \{0,1\}^{kl}$
$(x, x') \leftarrow\$ \mathcal{B}(\mathsf{bk}, \mathsf{H}, \mathsf{k}, x)$
return $(\mathsf{k}, x, x')$
```

Figure 4.1: Security games for hash functions with reusable backdoors

a challenger and the adversary $\mathcal{B}$ with backdoor key (from Definition 5). The transcript consists of all parameters and return values of the adversary $\mathcal{B}$ except the backdoor key.

We define the security games and advantages for second-preimage resistance and collision resistance. The same game transcript approach can be applied to preimage resistance or other cryptographic security notion.

Definition 8 (Second-Preimage Resistance with Reusable Backdoor)**.** A hash function generator BDHGen provides second-preimage resistance with reusable backdoor if BDHGen is SPR-backdoored hash function generator with adversary $\mathcal{B}$ (Definition 5) and for every PPT adversary $\mathcal{A}$ its advantage in the SPR game in Fig. 4.1 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{SPR},\mathcal{B}}_{\mathsf{BDHGen},\mathcal{A}}(\lambda) = \Pr\left[\mathrm{SPR}^{\mathcal{B}}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)\right]$$

The probability is over the choices of k and x and the random coins of $\mathcal{A}$ and $\mathcal{B}$.

Definition 9 (Collision Resistance with Reusable Backdoor)**.** A hash function generator BDHGen provides collision resistance with reusable backdoor if BDHGen is CR-backdoored hash function with an adversary $\mathcal{B}$ (Definition 5) and for every PPT adversary $\mathcal{A}$ its advantage in the CR game in Fig. 4.1 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{CR},\mathcal{B}}_{\mathsf{BDHGen},\mathcal{A}}(\lambda) = \Pr\left[\mathrm{CR}^{\mathcal{B}}_{\mathsf{BDHGen},\mathcal{A}}(1^\lambda)\right]$$

The probability is over the choices of k and the random coins of $\mathcal{A}$ and $\mathcal{B}$.

The original second-preimage resistance and collision resistance games are obtained by removing the oracle (and thus the adversary with backdoor key) in the respective game for reusable backdoors. The oracles do not answer with valid second-preimages or collisions every time, because we only assume non-negligible advantage from the adversary with backdoor key $\mathcal{B}$. If a valid answer is necessary the adversary $\mathcal{A}$ can query the oracle multiple times and receive a valid answer with high probability. As this property is needed in multiple later sections, it is proven as a lemma next.

Lemma 1. *Let $\mathcal{B}$ be a* PPT *adversary with input space $\mathcal{X}$ and output space $\mathcal{Y}$. Let $c : \mathcal{Y} \to \{0, 1\}$ be a polynomial time computable predicate. Assume $\mathcal{B}$ on input $x \leftarrow\$\, \mathcal{X}$ outputs a y such that $c(y) = 1$ with non-negligible probability in λ. Given n define the set of n inputs X, the set of n outputs Y and the set of valid outputs V as*

$$\begin{aligned} X &= \{x_i \mid x_i \leftarrow\$\, \mathcal{X} \wedge i \in \{1, \dots, n\}\} \\ Y &= \{y_i \mid y_i \leftarrow\$\, \mathcal{B}(x_i) \wedge x_i \in X\} \\ V &= \{y_i \mid c(y_i) = 1 \wedge y_i \in Y\} \end{aligned}$$

.

The probability that $|V| < q$ for some $q \in \mathbb{N}$ can be made exponentially small in n for infinitely many λ.

Proof. The probability of any $y_i \in Y$ being valid is non-negligible by definition and thus for infinitely many λ is bigger than a polynomial

$$\Pr[f(y_i) = 1] \geq \frac{1}{\mathsf{poly}(\lambda)} \tag{4.1}$$

For those λ and $\mathsf{poly}(\lambda)$ the probability that any y_i is invalid is then bounded by $1 - \frac{1}{\mathsf{poly}(\lambda)}$. We find a bound for any number q of elements in V. The Hoeffding-Chernoff bound implies that for $n = (q + m) * \mathsf{poly}(\lambda) * 4$ the probability that $|V| < q$ is exponentially small in $(q + m)$. Using the Chernoff bound $\Pr[|V| < (1 - \gamma)pn] \leq e^{-np\gamma^2 \frac{1}{2}}$ and setting $\gamma >= 1 - \frac{1}{4}$ and $n = (q + m) * \mathsf{poly}(\lambda) * 4$

$$\begin{aligned} \Pr[|V| < (1 - \gamma)pn] &\leq e^{-np\gamma^2 \frac{1}{2}} \\ \Pr[|V| < q + m] &\leq e^{-(q+m)\frac{9}{8}} \\ &\leq 2^{-(q+m)} \end{aligned}$$

□

We strive for a weak model of capabilities of adversary $\mathcal{A}$ against $\mathcal{B}$ that still allows us to argue implausibility later. We now discuss a few alternative ways to model reusable backdoors. One alternative is to let the adversary $\mathcal{A}$ act as a challenger in a game against $\mathcal{B}$. $\mathcal{A}$ is then able to query on a specified k (and x for second-preimage resistance). The BD oracle injects the backdoor key bk and returns $(x, x') \leftarrow\!\!\$\, \mathcal{B}(\mathsf{bk}, \mathsf{H}, k)$ ($x \leftarrow\!\!\$\, \mathcal{B}(\mathsf{bk}, \mathsf{H}, k, x)$ respectively). Now we can distinguish between adaptive and non-adaptive versions. In the adaptive version queries from $\mathcal{A}$ can depend on previous queries which is not the case in the non-adaptive version. This model is stronger because $\mathcal{A}$ can choose specific queries and get a targeted answer from $\mathcal{B}$. $\mathcal{A}$ can also randomly sample k (and x for second-preimage resistance) and emulate the model we chose. Another way is to fix k throughout the whole game. This means that the oracle BD does not choose a fresh key but uses the same that $\mathcal{A}$ is challenged on. Intuitively this should increase the chances of $\mathcal{A}$ in winning the game. We choose not to model a fixed key because at least for collision resistance this arguably models the wrong property. Under a fixed key, an adversary $\mathcal{A}$ only needs to query a single collision and thereafter output this collision as its own one, i.e., no hash function can be secure in this model. This impossibility can be avoided by requiring $\mathcal{A}$ to output a collision distinct from the collisions output by the backdoor oracle. This notion then essentially tests whether it is possible for an adversary to find further colliding elements to already known collisions. In the case of second-preimage resistance our model works without a key. The hash function can simply ignore the key and the game is still meaningful under this condition.

4.2 Implausibility of Backdoored Second-Preimage Resistant Functions

A hash function that is second-preimage resistant with reusable backdoor allows adversaries with backdoor key to find second-preimages under the hash function. Adversaries without backdoor key are unable to find second-preimages, but checking whether an input is a valid second-preimage under the hash function is easy. Using second-preimages as a signature allows adversaries with backdoor key to sign messages, while adversaries without backdoor key can only verify. We formalize this intuition and show that a EUF-CMA secure signature scheme can be constructed from this intuition in the random oracle model [BR93]. The notion of EUF-CMA allows the adversary to query signatures for messages of its choice. We match messages to oracle queries in the reusable backdoor games by pre-processing the message with a random oracle G (denoted as a uniformly random sampled function) and programming the random oracle in the reduction.

The scheme in Fig. 4.2 is infinitely often correct assuming the advantage of adversary with backdoor key $\mathcal{B}$ is non-negligible, i.e.,

$$\mathsf{Adv}^{\mathrm{SPR},1}_{\mathsf{BDHGen},\mathcal{B}}(\lambda) \geq \frac{1}{\mathsf{poly}(\lambda)}$$

for infinitely many λ.

```
$\Pi.\mathsf{KGen}(1^\lambda)$
    $(\mathsf{H}, \mathsf{bk}) \leftarrow\!\$\, \mathsf{BDHGen}(1^\lambda)$
    $\mathsf{G} \leftarrow\!\$\, \mathsf{Fun}[*, il]$
    return $(\mathsf{vk} = \mathsf{H}, \mathsf{sk} = \mathsf{bk})$
```

```
$\Pi.\mathsf{Sig}(\mathsf{bk}, m)$
    $\mathsf{k} \leftarrow\!\$\, \{0,1\}^\lambda$
    $n \leftarrow\!\$\, \{0,1\}^\lambda$
    $x = \mathsf{G}(n \| m)$
    $\sigma \leftarrow\!\$\, \mathcal{B}(\mathsf{bk}, \mathsf{H}, \mathsf{k}, x)$
    return $(\mathsf{k}, n, \sigma)$
```

```
$\Pi.\mathsf{Vf}(\mathsf{H}, k, n, m, \sigma)$
    $x' = \mathsf{G}(n \| m)$
    return $\mathsf{H}_\mathsf{k}(\sigma) = \mathsf{H}_\mathsf{k}(x) \wedge \sigma \neq x$
```

Figure 4.2: Digital Signatures from second-preimage resistant functions with reusable backdoor

Correctness: The signature is valid if $\mathcal{B}$ is able to find a second-preimage for H. $\mathcal{B}$ assumes that x is sampled uniformly random. The distribution of x is uniform random because G is modelled as a random oracle. The probability of obtaining a valid signature from $\mathcal{B}$ is then the advantage of adversary with backdoor key $\mathcal{B}$.

$$\Pr[\mathsf{Vf}(\mathsf{vk}, m, (\mathsf{k}, \sigma))] = \mathsf{Adv}^{\mathrm{SPR},1}_{\mathsf{BDHGen},\mathcal{B}}(\lambda) \geq \frac{1}{\mathsf{poly}(\lambda)}$$

G does not need to be a random oracle for correctness to work. If G is a PRF and the adversary $\mathcal{B}$ never learns the PRF key, the same argument as above can be applied for correctness. The following security reduction does still need the random oracle model.

Theorem 1. *Let* BDHGen *be a hash function generator,* $\mathsf{H} : \{0,1\}^{kl} \times \{0,1\}^{il} \mapsto \{0,1\}^{ol}$ *a backdoored second-preimage resistant hash function. Let* $\mathsf{G} : \{0,1\}^* \to \{0,1\}^{il}$ *be a random oracle. The construction in Fig. 4.2 is an infinitely often* EUF-CMA *secure signature scheme with advantage*

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Pi,\mathcal{C}}(\lambda) \leq (q_s + q_r)\mathsf{Adv}^{\mathrm{SPR},\mathcal{B}}_{\mathsf{BDHGen},\mathcal{D}}(\lambda)$$

where q_r *is the number of random oracle queries of* $\mathcal{C}$ *and* q_s *is the number of signatures queries of* $\mathcal{C}$*.*

Proof. We prove EUF-CMA security by reducing to second-preimage resistance with reusable backdoor of H. Assume an adversary $\mathcal{C}$ that violates the EUF-CMA security of the scheme and does at most q_s signature queries and at most q_r random oracle queries. We build an adversary $\mathcal{D}$ that violates second-preimage resistance with reusable backdoor of H by using $\mathcal{C}$.

When $\mathcal{C}$ queries the random oracle, $\mathcal{D}$ queries the backdoor oracle SPR-BD and gets an answer consisting of a uniform random hash key k and two messages x, x' that collide under the given hash key with non-negligible probability. $\mathcal{C}$ receives x as a response for the random oracle. When $\mathcal{C}$ queries for a signature of message x, $\mathcal{D}$ generates a nonce n and queries the random oracle $n \| x$ as though $\mathcal{C}$ would query the random oracle and x'

(i.e., the second part of the colliding pair) is returned as the signature. $\mathcal{D}$ embeds the preimage it receives from the second-preimage resistance challenger as one of the random oracle answers at random. Assuming $\mathcal{C}$ makes at most $q_s + q_r$ random oracle queries including the implicit queries via the signing oracle, the probability that $\mathcal{D}$ embeds its second-preimage resistance challenge in the query that $\mathcal{C}$ uses for its forgery is at least $\frac{1}{q_s+q_r}$, thus yielding the bound.

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Pi,\mathcal{C}}(\lambda) \leq (q_s + q_r)\mathsf{Adv}^{\text{SPR},\mathcal{B}}_{\mathsf{BDHGen},\mathcal{D}}(\lambda)$$

□

We can now amplify correctness of the signatures scheme. The success probability of valid signatures can be amplified by running $\mathcal{B}$ multiple times and applying Lemma 1 with a big enough number of repetitions n. Define $q = 1$, $m = \lambda$, $\mathcal{X}$ as the possible inputs to $\mathcal{B}$, $\mathcal{Y}$ as the possible outputs of $\mathcal{B}$ and c as the function checking whether the output of $\mathcal{B}$ is a valid second preimage. In order to ensure independent samples, we need n independent backdoor keys. The signature key is then a vector of n backdoor keys and the verification key needs to contain all n hash function descriptions. The verification algorithm is either provided a reference to the correct hash function description by the signature or needs to try verification with every hash function description. We recommend a reference in the signature to not increase verification time. The correctness error is then exponentially small for infinitely many λ by Lemma 1.

4.3 Implausibility of Backdoored Collision Resistant Functions

We show a similar result for collision resistance with reusable backdoors. A reusable backdoor in the collision resistance notion of a hash function allows an adversary with backdoor key to find colliding inputs under a specified key, while adversaries without backdoor key are unable to do so. Verifying that two inputs collide under a given key is easy. Using the colliding pair as a signature for a message used as the key allows us to build a signature scheme from a hash function with collision resistance with reusable backdoors. Note that for a compression function to be collision resistant in the common notion of collision resistance, it needs to be keyed. We use a random oracle again to pre-process the message and generate a random key. The key space is defined as $\{0,1\}^{kl}$, i.e., all bit strings of certain length are valid keys.

Correctness: The signature is valid if $\mathcal{B}$ is able to find a collision for H under k. $\mathcal{B}$ assumes that k is sampled uniformly random. The distribution of k is uniform random because we model G as a random oracle. The probability of obtaining a valid signature from $\mathcal{B}$ is the advantage of the adversary with backdoor key.

$$\Pr[\mathsf{Vf}(\mathsf{vk}, m, (\mathsf{k}, \sigma))] = \mathsf{Adv}^{\text{CR},1}_{\mathsf{BDHGen},\mathcal{B}}(\lambda)$$

```
$\Pi.\mathsf{KGen}(1^\lambda)$
    $(\mathsf{H}, \mathsf{bk}) \leftarrow\$ \mathsf{BDHGen}(1^\lambda)$
    $\mathsf{G} \leftarrow\$ \mathsf{Fun}[*, kl]$
    return $(\mathsf{vk} = \mathsf{H}, \mathsf{sk} = \mathsf{bk})$
```

```
$\Pi.\mathsf{Sig}(\mathsf{bk}, m)$
    $n \leftarrow\$ \{0,1\}^\lambda$
    $\mathsf{k} = \mathsf{G}(n \| m)$
    $(x, x') \leftarrow\$ \mathcal{B}(\mathsf{bk}, \mathsf{H}, \mathsf{k})$
    return $(\mathsf{k}, n, (x, x'))$
```

```
$\Pi.\mathsf{Vf}(\mathsf{H}, k, n, m, (x, x'))$
    $\mathsf{k} = \mathsf{G}(n \| m)$
    return $\mathsf{H}_\mathsf{k}(x) = \mathsf{H}_\mathsf{k}(x') \land x \neq x'$
```

Figure 4.3: Digital Signatures from collision resistant functions with reusable backdoor

Theorem 2. *Let* BDHGen *be a hash function generator,* $\mathsf{H} : \{0,1\}^{kl} \times \{0,1\}^{il} \mapsto \{0,1\}^{ol}$ *a backdoored hash function. Let* $\mathsf{G} : \{0,1\}^* \to \{0,1\}^{kl}$ *be a random oracle. The construction in Fig. 4.3 is an infinitely often* EUF-CMA *secure signature scheme with advantage*

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Pi,\mathcal{C}}(\lambda) \leq (q_s + q_r)\mathsf{Adv}^{\text{CR},\mathcal{B}}_{\mathsf{BDHGen},\mathcal{D}}(\lambda)$$

where q_r *is the number of random oracle queries of* $\mathcal{C}$ *and* q_s *is the number of signature oracle queries of* $\mathcal{C}$*.*

Proof. We prove EUF-CMA security by reducing to the collision resistance with reusable backdoor of H. Assume an adversary $\mathcal{C}$ that violates the EUF-CMA security of the scheme and does at most q_r signature queries and at most q_r random oracle queries. We build an adversary $\mathcal{D}$ that violates collision resistance with reusable backdoor of H by using $\mathcal{C}$.

When $\mathcal{C}$ queries the random oracle, $\mathcal{D}$ queries the backdoor oracle CR-BD and gets an answer consisting of a uniform random hash key k and two messages x, x' that collide under the given hash key with non-negligible probability. $\mathcal{C}$ receives k as a response for the random oracle. When $\mathcal{C}$ queries for a signature of message m, $\mathcal{D}$ samples a nonce n and queries the random oracle for $n \| m$ as though $\mathcal{C}$ would query the random oracle. The collision (x, x') returned by the backdoor oracle is returned as the signature. $\mathcal{D}$ embeds the collision challenge it receives from the collision resistance challenger in the form of a target key as one of the random oracle answers at random. Assuming $\mathcal{C}$ makes at most $q_s + q_r$ random oracle queries (including the random oracle queries that are implicitly done by signature queries), the probability that $\mathcal{D}$ embeds its collision resistance challenge in the query that $\mathcal{C}$ uses for its forgery is at least $\frac{1}{q_s+q_r}$, thus yielding the bound.

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Pi,\mathcal{C}}(\lambda) \leq (q_s + q_r)\mathsf{Adv}^{\text{CR},\mathcal{B}}_{\mathsf{BDHGen},\mathcal{D}}(\lambda)$$

□

Correctness of this signatures scheme can be amplified by running $\mathcal{B}$ multiple times and applying Lemma 1 with a big enough number of repetitions n, similar to the second-preimage resistance case. Define $q = 1$, $m = \lambda$, $\mathcal{X}$ as the possible inputs to $\mathcal{B}$, $\mathcal{Y}$ as the possible outputs of $\mathcal{B}$ and c as the function checking whether the output of $\mathcal{B}$ is a valid collision. The correctness error is then exponentially small for infinitely many λ by Lemma 1.

Any collision resistant function is also second-preimage resistant [RS04]. The above result thus establishes another way to get second-preimage resistance with implausibility of backdoors.

Collision resistant hash functions are a standard notion that is assumed by many signature schemes and other higher level cryptographic primitives and protocols. Assuming reusable backdoors allows us to state implausibility of backdoors in (efficient) collision resistant compression functions.

4.4 Interpretation of Results

While the above results can in principle be applied to any hash functions that is assumed to be second-preimage resistant or collision resistant, they are meaningful for compression functions (instead of full hash functions) with small and fixed length input spaces.

A compression function with a reusable backdoor would entail signature schemes with small signatures sizes and fast verification times. In the second-preimage resistance case, the signature is a hash key, a nonce and a single element from the input space. The collision resistance case needs a hash key, a nonce and two elements from the input space (the collision). If the instantiation for the random oracle G is deterministic the hash key can even be left out of the signature. Verification time for both constructions is dominated by the random oracle call and two compression function evaluations. Performance of key generation and signature generation depends on the algorithms of BDHGen and $\mathcal{B}$. An efficient and highly correct adversary with backdoor key is valuable for malicious designers attacking time critical applications that use the compression function. For the above constructions such an adversary with backdoor key means fast signature generation as this algorithm is most likely the dominating factor.

Correctness and security of the both schemes does not hold unconditionally but rather only infinitely often. Both whether the scheme is correct and secure for a specific security parameter λ depends on the advantage of the adversary with backdoor key against this security parameter. If the advantage is high against a concrete value of λ then the schemes are secure and correct for the same value of λ. An adversary designing a backdoored hash function would need the backdoor to work for parameters used in practice, which means that for those parameters both schemes are then correct and secure. Infinitely often correctness and security is thus not a serious limitations.

In previous similar implausibility results [FJM18] and the one we will reprove in Chapter 5, the backdoored primitive, a PRF and a PRG, belong to a different class of cryptographic primitives than the implied public key encryption. Both PRF and PRG are implied by preimage resistant functions [Hås+99; GGM86] and vice versa while public key encryption is not implied by preimage resistant functions in a black-box way [IR89]. In our case, second-preimage resistance and signature schemes are both implied by preimage resistant functions [Rom90; KK05] and vice versa [RS04]. They belong to the same class of primitives and thus the argument that backdoored second-preimage resistant (or collision resistant) functions need to contain enough structure to build signature schemes, because they are easily constructed by second-preimage resistant (or collision resistant) functions, does not tell us much. Depending on the exact reusable backdoor parameters, the proposed signature schemes may well be very efficient and have much smaller signature sizes than comparable hash-based signature schemes. Note that our proposed constructions allow a virtually infinite number of signatures under a single verification key and are thus comparable to (hash-based) Merkle signature schemes like XMSS or SPHINCS instead of (hash-based) few-time signature schemes like WOTS and HORS. An efficient compression function with a reusable backdoor is then still unlikely without major improvements in the efficiency and signature size of (many-time) signature schemes.

CHAPTER 5

Few-Time Signatures with Implausibility of Backdoors

The first step to achieving hash-based signature schemes with implausibility of backdoors is building few-time signature schemes with implausibility of backdoors.

Among the few-time signature schemes considered, HORS++ [PWX04] is best suited for our standard model endeavour because it only needs preimage resistance to prove security.

We immunize preimage resistance against backdoors by showing that pseudorandom generators have implausibility of backdoors. Pseudorandom generators that are expanding already are preimage resistant and thus are preimage resistance with implausibility of backdoors. Practical examples of functions assumed to be PRGs are part of stream ciphers such as Salsa [Ber08]. More specifically, using the pseudorandom keystream as an output directly or encrypting the all zero message yields a PRG.

A PRG that expands by exactly one bit will be the most efficient choice. Fortunately, every PRG that expands by more can be truncated. If there is an adversary that can distinguish the truncated output of a PRG from a uniform random string, this adversary can be used in an adversary that distinguishes the original PRG.

5.1 Modelling Backdoored Pseudorandom Generators

A backdoored PRG is a PRG (with the usual security notion [KL14]), where there exists an adversary that can, equipped with a backdoor key bk, violate the pseudorandomness notion of the PRG. Our notion of a backdoored PRG is adapted from the notion of a (strong) backdoored PRF [FJM18]. We only need the strong notion, where the generator is not allowed to influence the randomness used during key generation.

```
$\mathrm{PR}^{bd}_{\mathsf{BDPRGGen},\mathcal{A}}(1^\lambda)$

$(\mathsf{G}, \mathsf{bk}) \leftarrow\!\$\ \mathsf{BDPRGGen}(1^\lambda)$
if $bd = 0$ then $\mathsf{bk} = \bot$
$b \leftarrow\!\$\ \{0,1\}$
$x \leftarrow\!\$\ \{0,1\}^{il}$
if $b = 1$ then
    $y = \mathsf{G}(x)$
else
    $y \leftarrow\!\$\ \{0,1\}^{ol}$
$b' \leftarrow\!\$\ \mathcal{A}(\mathsf{bk}, \mathsf{G}, \mathsf{k}, y)$
return $b = b'$
```

Figure 5.1: Security game for pseudorandom generators

Definition 10 (PRG generator)**.** A PPT algorithm BDPRGGen is called a PRG generator if on input a security parameter 1^λ, it outputs a PRG $\mathsf{G} : \{0,1\}^{il} \to \{0,1\}^{ol}$ and a backdoor key bk, for length parameters $il, ol \in \mathbb{N}$ with $il < ol$ and depending on the security parameter λ. G is called a pseudorandom generator or PRG.

The usual length extension property of a PRG remains unchanged in the presence of backdoors, but we do need to adapt the pseudorandomness notion. Similarly to the hash function games above, setting $bd = 0$ yields the standard pseudorandomness definition of a PRG.

Definition 11 (Pseudorandomness of a PRG)**.** A generator G provides pseudorandomness (with backdoor for $bd = 1$) if for every PPT adversary $\mathcal{A}$ its advantage in the PR game in Fig. 5.1 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{PR},bd}_{\mathsf{BDPRGGen},\mathcal{A}}(\lambda) = 2 \cdot \Pr\left[\mathrm{PR}^{bd}_{\mathsf{BDPRGGen},\mathcal{A}}(1^\lambda)\right] - 1$$

The probability is over the choice of b, x and y and the random coins of $\mathcal{A}$.

5.2 Implausibility of Backdoored Pseudorandom Generators

A backdoored PRG allows an adversary holding the backdoor key bk to violate the security notion of the PRG, i.e., distinguish PRG outputs from uniform random strings. Choosing the PRG output as an encryption of $b = 1$ and a uniform random string for $b = 0$ allows us to use this adversary to decrypt to b, thus a backdoored PRG implies bit encryption. Although this idea is not new, we choose a different formalisation of

```
$\Pi.\mathsf{KGen}(1^\lambda)$
    $(\mathsf{G}, \mathsf{bk}) \leftarrow\!\!\$\ \mathsf{BDPRGGen}(1^\lambda)$
    return $(\mathsf{pk} = \mathsf{G}, \mathsf{sk} = \mathsf{bk})$
```

```
$\Pi.\mathsf{Enc}(\mathsf{G}, b)$
    $\mathsf{k} \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$
    $d \leftarrow\!\!\$\ \{0,1\}$
    $b' = d \oplus b$
    if $b' = 0$ then
        $c \leftarrow\!\!\$\ \{0,1\}^{l(n)}$
    else
        $s \leftarrow\!\!\$\ \{0,1\}^n$
        $c = \mathsf{G}(s)$
    return $(d, c)$
```

```
$\Pi.\mathsf{Dec}(\mathsf{bk}, (d, c))$
    $b' = \mathcal{A}(\mathsf{bk}, c, \mathsf{G})$
    $b = b' \oplus d$
    return $b$
```

Figure 5.2: Bit-encryption scheme from backdoored pseudorandom generators

PRGs than previous work [Dod+15] and fix a subtle error in the construction. The difference in formalisation is that we use a stateless definition of a PRG instead of a definition that iterates the PRG on a state to get more output. The subtle error is the following: Correctness of the original construction [Dod+15] cannot be amplified if the adversary against the backdoored PRG behaves in a particular one-sided way. The same subtlety was remarked in a similar construction of bit-encryption from backdoored weak PRFs [FJM18]. For example, if an adversary outputs 1 with probability 1 when given a random string and outputs 1 with probability $1 - \varepsilon$ for a noticeable ε for a pseudorandom string, amplification through parallel repetition and majority vote does not work. This can be fixed by blinding the bit before encrypting it. We therefore modify the bit encryption such that the bit is blinded before encrypting.

Figure 5.2 shows the bit-encryption scheme. The scheme is infinitely often correct assuming the advantage of the adversary is non-negligible, i.e.,

$$\mathsf{Adv}^{\mathrm{PR},1}_{\mathsf{BDPRGGen},\mathcal{A}}(\lambda) \geq \frac{1}{\mathsf{poly}(\lambda)}$$

for infinitely many λ.

Correctness: The ciphertext can be correctly decrypted if $\mathcal{A}$ correctly distinguishes the pseudorandom output from a uniform random string. We assume that the advantage of the adversary with backdoor key is non-negligible.

$$\mathsf{Adv}^{\mathrm{PR},1}_{\mathsf{BDPRGGen},\mathcal{A}}(\lambda) = \varepsilon \geq \frac{1}{\mathsf{poly}(\lambda)}$$

Thus for infinitely many λ, the probability of a successful decryption is

$$\begin{aligned}\Pr[\mathsf{Dec}(\mathsf{bk}, \mathsf{Enc}(\mathsf{G}, b) = b] &= \Pr[\mathcal{A}(\mathsf{bk}, \mathsf{Enc}(\mathsf{G}, b), \mathsf{G}) = b] \\ &= \frac{1 + \mathsf{Adv}^{\mathrm{PR},1}_{\mathsf{BDPRGGen},\mathcal{A}}(\lambda)}{2} \\ &= \frac{1}{2} + \frac{\varepsilon}{2}.\end{aligned}$$

Adversaries that do not hold the backdoor key cannot distinguish pseudorandom outputs of G from uniform random samples in the output space. The bit-encryption scheme is thus IND-CPA secure.

Theorem 3. *Let* $\mathsf{BDPRGGen}$ *be a PRG generator and* $\mathsf{G} : \{0,1\}^{i(n)} \mapsto \{0,1\}^{o(n)}$ *be a backdoored PRG and* $\mathcal{A}$ *a* PPT *adversary against* G*. The construction in Fig. 5.2 is an* IND-CPA *secure bit-encryption scheme with advantage:*

$$\mathsf{Adv}^{\mathrm{IND\text{-}CPA}}_{\mathsf{BPRG},\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathrm{PR},0}_{\mathsf{BDPRGGen},\mathcal{C}}(\lambda)$$

Proof. We proof IND-CPA security by reducing to pseudorandomness of G against adversaries without the backdoor key bk. Assume an adversary $\mathcal{B}$ that violates the IND-CPA security of the scheme. We use $\mathcal{B}$ to build an adversary $\mathcal{C}$ that distinguishes the output of G from a uniform random string, even without the backdoor key bk. $\mathcal{C}$ plays the pseudorandomness game and receives the description of the PRG G and a challenge y which is either a random string or an output of G. $\mathcal{C}$ samples a random bit d and gives (d, y) to $\mathcal{B}$. When $\mathcal{B}$ returns the message bit m, $\mathcal{C}$ computes $m \oplus d$ and outputs this as its guess. The advantage of $\mathcal{C}$ in the pseudorandomness game is the same as the advantage of $\mathcal{C}$ in the IND-CPA game.

$$\mathsf{Adv}^{\mathrm{IND\text{-}CPA}}_{\mathsf{BPRG},\mathcal{B}}(\lambda) = \mathsf{Adv}^{\mathrm{PR},0}_{\mathsf{BDPRGGen},\mathcal{C}}(\lambda)$$

□

We can amplify successful decryption probability by repeating the encryption step a polynomial number of times. Decryption then works by a majority decision on all the decryption results. With success probability $\varepsilon \geq \frac{1}{\mathsf{poly}(\lambda)}$ for every repetition, the Hoeffding-Chernoff bound implies that for $\lambda \cdot \mathsf{poly}(\lambda)^2$ repetitions ($\lambda \cdot \mathsf{poly}(\lambda)^2 + 1$ if $\lambda \cdot \mathsf{poly}(\lambda)^2$ is even) the decryption error is smaller then $\mathrm{e}^{-\lambda}$.

A PRG can be built from unkeyed preimage resistant functions (also called one-way functions) [Hås+99] and vice versa, while public key encryption is not implied by unkeyed preimage resistant functions [IR89]. A backdoored PRG constructions thus needs to contain enough structure in itself to allow for public key encryption. This structure is likely detectable by analysing the design of the PRG. Practical PRGs are faster than any known public key encryption, suggesting that instantiations of a backdoored PRG are either suspiciously inefficient or provide major efficiency improvements for public key encryption. The case of $\mathsf{DUAL\_EC\_DRBG}$ is a good example [BLN15]. Choosing parameters for $\mathsf{DUAL\_EC\_DRBG}$ allows one to include a backdoor into the PRG. The design does look suspicious because its structure is reminiscent of public key encryption schemes and $\mathsf{DUAL\_EC\_DRBG}$ itself is much slower then other common PRGs. We refer to the referenced work for details of the history of $\mathsf{DUAL\_EC\_DRBG}$. Note that we show implausibility not impossibility. Advancements in the efficiency of public key encryption are possible and it may be possible to embed a public key encryption structure into the PRG that is hard to detect, e.g., a scheme the public does not know about or an obfuscated one.

5.3 Immunizing Few-Time Signatures in the Standard Model

Recall that if G is an expanding PRG then $f(x) = \mathsf{G}(x)$ is preimage resistant. It is then implausible to embed a backdoor into f that allows an adversary to violate preimage resistance of f. We will now use f as a preimage resistant function in $\mathsf{HORS{+}{+}}$. This allows us to construct a few-time signature scheme with implausibility of backdoors.

The general idea of HORS variants [RR02; PWX04] is to generate a set of secret bit strings as the signature key. The corresponding verification key is the set of images of a preimage resistant function f applied to the signature key. In order to sign a message m a specific subset of the signature key is chosen by a selection function S and revealed as the signature for m. The verification algorithm can apply f to the signature subset and check whether the images are contained in the verification key and the correct subset is used in the signature. The signature scheme can be used for n signatures if for any $n+1$ different messages $m_1, \dots, m_{n+1}$, the first n subsets chosen by S $S(m_1), \dots, S(m_n)$ are missing at least one element needed for $S(m_{n+1})$. If this property is violated, an adversary can request n signatures and construct a new signature out of the revealed signature key elements. An important choice and the main difference between variants of HORS is the selection function S. HORS uses a hash function as S and interprets substrings of the output as indices of the signature key to include in the signature. $\mathsf{HORS{+}{+}}$ uses cover-free families to construct an S that guarantees the above property.

Definition 12. A set system (U, S) with $U = \{u_1, \dots u_t\}$ and $S = \{S_i \subseteq U \mid i = 1, \dots, n\}$ is a (n, t, r)-cover-free family if for any subset $T \subseteq S$ of size r and any $S_i \notin T$

$$\left| S_i \setminus \bigcup T \right| \geq 1$$

The property of the cover-free family ensures that up to r different messages can be signed and the adversary is still forced to find the preimage of at least one element of the verification key. Cover-free families exist for reasonable $\mathsf{HORS{+}{+}}$ parameters allowing signature schemes with a few kilobytes signature and key size. We refer to the original work for details [PWX04].

5.4 Immunizing Few-Time Signatures in the Random Oracle Model

$\mathsf{HORS{+}{+}}$ is not as efficient as one-time signatures used in modern hash-based signatures, e.g., WOTS^+ used in XMSS and $\mathsf{XMSS\text{-}T}$. WOTS^+ needs a function that simultaneously provides preimage resistance, second-preimage resistance and pseudorandomness. In order to get closer in efficiency to practical hash-based signatures we would like to construct such a function with implausibility of backdoors for all three requirements. We did not manage to construct one in the standard model, much less an efficient construction for it. If we relax the restrictions of the standard model, we are able to construct such a function. It has already been shown that combining multiple backdoored random oracles can provide preimage resistance and weak pseudorandomness [BFM18]. The backdoors are modeled as a backdoor oracle for each individual random oracle H that on input a function f returns the evaluation of f on the random oracle function table. The function f itself is unrestricted and the queries can be adaptive.

Definition 13 (The BRO Model)**.** The backdoored random oracle (BRO) model equips a random oracle $\mathsf{H} \leftarrow\!\!\$\, \mathsf{Fun}[il, ol]$ with a backdoor oracle BD that is accessible to all adversaries. The oracle BD is queried with a function $f \in \mathcal{F}$ and returns the result of f applied to the function table of H.

$$\mathsf{BD}(f) = f(\langle \mathsf{H} \rangle)$$

If $\mathcal{F} = \emptyset$ the BRO model becomes the conventional random oracle model.

The k-BRO model is defined by extending the BRO model to k independent functions $\mathsf{H}_1, \ldots \mathsf{H}_k$. Adversaries have access to each of the k independent backdoor oracles $\mathsf{BD}_i(f) = f(\langle \mathsf{H}_i \rangle)$ for $i \in 1, \ldots k$.

We assume PPT adversaries, implicitly restricting $\mathcal{F}$ to functions with polynomial size descriptions. The security reduction of the combiner does not need this restriction. As we only restrict the classes of functions the adversary can use in its backdoor oracles, the results for preimage resistance and weak pseudorandomness still hold under this restriction.

WOTS^+ inputs are of constant size, thus we can model a single compression function as a backdoored random oracle instead of the whole hash function with arbitrary length input. We now present the definitions of preimage resistance and image uniformity [BFM18] and our adaption to second-preimage resistance of backdoored random oracle combiners and then argue why each of the requirements for a hash function in WOTS^+ are met by the 2-BRO concatenation combiner $C^{\mathsf{H}_1,\mathsf{H}_2}(x) = \mathsf{H}_1(x) \| \mathsf{H}_2(x)$.

Definition 14 (Preimage Resistance (BRO))**.** A random oracle combiner $C^{\mathsf{H}_1,\mathsf{H}_2}$ provides preimage resistance if for every PPT adversary $\mathcal{A}$ its advantage in the PRE game in

```
$\mathrm{PRE}_{\mathsf{C},\mathcal{A}}(1^\lambda)$

$\mathsf{H}_1 \leftarrow\$ \mathsf{Fun}[il, ol]$
$\mathsf{H}_2 \leftarrow\$ \mathsf{Fun}[il, ol]$
$x \leftarrow\$ \{0,1\}^{il}$
$y = C^{\mathsf{H}_1,\mathsf{H}_2}(x)$
$x' \leftarrow\$ \mathcal{A}^{\mathsf{H}_1,\mathsf{H}_2,\mathsf{BD}_1,\mathsf{BD}_2}(y)$
return $C^{\mathsf{H}_1,\mathsf{H}_2}(x') = y$
```

```
$\mathrm{SPR}_{\mathsf{C},\mathcal{A}}(1^\lambda)$

$\mathsf{H}_1 \leftarrow\$ \mathsf{Fun}[il, ol]$
$\mathsf{H}_2 \leftarrow\$ \mathsf{Fun}[il, ol]$
$x \leftarrow\$ \{0,1\}^{il}$
$x' \leftarrow\$ \mathcal{A}^{\mathsf{H}_1,\mathsf{H}_2,\mathsf{BD}_1,\mathsf{BD}_2}(x)$
return $C^{\mathsf{H}_1,\mathsf{H}_2}(x') = C^{\mathsf{H}_1,\mathsf{H}_2}(x)$
        $\wedge\, x \neq x'$
```

```
$\mathrm{IU}_{\mathsf{C},\mathcal{A}}(1^\lambda)$

$\mathsf{H}_1 \leftarrow\$ \mathsf{Fun}[il, ol]$
$\mathsf{H}_2 \leftarrow\$ \mathsf{Fun}[il, ol]$
$b \leftarrow\$ \{0,1\}$
$y_0 \leftarrow\$ \mathsf{Img}(C^{\mathsf{H}_1,\mathsf{H}_2})$
$x \leftarrow\$ \{0,1\}^{il}$
$y_1 = C^{\mathsf{H}_1,\mathsf{H}_2}(x)$
$b' \leftarrow\$ \mathcal{A}^{\mathsf{H}_1,\mathsf{H}_2,\mathsf{BD}_1,\mathsf{BD}_2}(y_b)$
return $b = b'$
```

Figure 5.3: Games for preimage resistance, second-preimage resistance and image uniformity with backdoored random oracles

Fig. 5.3 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{PRE}}_{C,\mathcal{A}}(\lambda) = \Pr\left[\mathrm{PRE}_{C,\mathcal{A}}(1^\lambda)\right]$$

The probability is over the choice of H_1, H_2, x and the random coins of $\mathcal{A}$.

Definition 15 (Second-Preimage Resistance (BRO))**.** A random oracle combiner $C^{\mathsf{H}_1,\mathsf{H}_2}$ provides second-preimage resistance if for every PPT adversary $\mathcal{A}$ its advantage in the SPR game in Fig. 5.3 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{SPR}}_{C,\mathcal{A}}(\lambda) = \Pr\left[\mathrm{SPR}_{C,\mathcal{A}}(1^\lambda)\right]$$

The probability is over the choice of H_1, H_2 and x and the random coins of $\mathcal{A}$.

Definition 16 (Image Uniformity (BRO))**.** A random oracle combiner $C^{\mathsf{H}_1,\mathsf{H}_2}$ provides image uniformity if for every PPT adversary $\mathcal{A}$ its advantage in the IU game in Fig. 5.3 is negligible in the security parameter λ.

$$\mathsf{Adv}^{\mathrm{IU}}_{C,\mathcal{A}}(\lambda) = 2 \cdot \Pr\left[\mathrm{IU}_{C,\mathcal{A}}(1^\lambda)\right] - 1$$

The probability is over the choice of H_1, H_2, b, y_0 and x and the random coins of $\mathcal{A}$.

Preimage Resistance of the Concatenation Combiner. For a range of compressing parameters the concatenation combiner of two backdoored random oracles is preimage resistant, even if the adversary can query both backdoor oracles [BFM18]. The backdoored random oracle version of the game is defined in Fig. 5.3 This fulfills our first requirement.

Pseudorandomness of the Concatenation Combiner. Image uniformity holds for the parameters that are valid for preimage resistance. The image uniformity game in Fig. 5.3 asks the adversary to distinguish a random sample from the image space of the combiner from an output of the combiner on a random input. What we need for pseudorandomness is that a random image is indistinguishable from a random co-domain point. The original paper also shows that for the compressing parameters used here, the probability that a random co-domain element is in the images space of $C^{\mathsf{H}_1,\mathsf{H}_2}$ is overwhelming, thus the probability that a random co-domain element is not in the image space is negligible. This fact implies that the image uniformity game is indistinguishable from the pseudorandomness game for compressing parameters. As image uniformity holds, the concatenation combiner provides pseudorandomness as well, fulfilling our second requirement.

Second-Preimage Resistance of the Concatenation Combiner. Assuming preimage resistance, we will show in the following that the concatenation combiner is also second-preimage resistant. Figure 5.3 shows the preimage resistance and second-preimage resistance games we use. The proof works by embedding the preimage challenge into both random oracles. Every function f that is sent to a backdoor oracle is composed with a function g that changes the function table to embed the challenge.

Theorem 4. *Let $C^{\mathsf{H}_1,\mathsf{H}_2} \in \mathsf{Fun}[il, ol]$ be the concatenation combiner in the* 2-BRO *model. For every* PPT*adversary $\mathcal{A}$ against the preimage resistance of $C^{\mathsf{H}_1,\mathsf{H}_2}$, there exists an adversary $\mathcal{B}$ against the second-preimage resistance of $C^{\mathsf{H}_1,\mathsf{H}_2}$ in the* 2-BRO *model.*

$$\mathsf{Adv}^{\mathrm{SPR}}_{C^{\mathsf{H}_1,\mathsf{H}_2},\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathrm{PRE}}_{C^{\mathsf{H}_1,\mathsf{H}_2},\mathcal{A}}(\lambda)$$

Proof. We assume an adversary $\mathcal{A}$ that violates second-preimage resistance of $C^{\mathsf{H}_1,\mathsf{H}_2}$ and use $\mathcal{A}$ to build an adversary $\mathcal{B}$ against preimage resistance of $C^{\mathsf{H}_1,\mathsf{H}_2}$. $\mathcal{B}$ first receives a challenge value $y \in \{0,1\}^m$, where $y = y_1 \| y_2$, y_1 is the output of H_1 and y_2 is the output of H_2. $\mathcal{B}$ needs to find a preimage x' for $C^{\mathsf{H}_1,\mathsf{H}_2}(x') = y$.

$\mathcal{B}$ samples a uniform random $x \leftarrow\$ \{0,1\}^n$ and gives this x as a challenge to $\mathcal{A}$. The main idea of $\mathcal{B}$ is to do one change to the two random functions H_1 and H_2 each such that $\mathsf{H}'_i(x) = y_i$. We assume without loss of generality that $\mathcal{A}$ only queries its backdoor oracles BD_i as queries to H_i can be implemented by queries to the respective backdoor oracle BD_i. We define a function δ_i that takes a function table of H_i and returns a function table that is the same except that $\mathsf{H}_i(x) = y_i$. When $\mathcal{B}$ receives a query f for BD_i it composes f with δ_i as $f' = f \circ \delta_i$. It then uses its own backdoor oracle to query f' and gives the answer to $\mathcal{A}$. We need to ensure that $f' \in \mathcal{F}$, i.e., the description f' is polynomial sized. The description of f is polynomial sized, because we assumed $\mathcal{A}$ to be a PPT adversary and its queries can only be of polynomial size. δ and f may not be computable in polynomial time, but composing $f \circ \delta$ (without evaluating) can be done in polynomial time. Via this method, $\mathcal{B}$ can answer all backdoor queries from $\mathcal{A}$. When $\mathcal{B}$ receives x' from $\mathcal{A}$ it returns x' as its preimage. If x' is a valid answer to the second-preimage resistance game it is a valid preimage to y under $C^{\mathsf{H}_1,\mathsf{H}_2}$. Thus the advantage of $\mathcal{B}$ against the preimage resistance game is equal to the advantage of $\mathcal{A}$ against the second-preimage resistance game.

$$\mathsf{Adv}^{\mathrm{SPR}}_{C^{\mathsf{H}_1,\mathsf{H}_2},\mathcal{B}}(\lambda) = \mathsf{Adv}^{\mathrm{PRE}}_{C^{\mathsf{H}_1,\mathsf{H}_2},\mathcal{A}}(\lambda)$$

□

The concatenation combiner of backdoored random oracles is preimage resistant, second-preimage resistant and pseudorandom. This idealized construction can be instantiated by two different (and independently designed) compression functions and used in WOTS^+, adapting the original WOTS^+ security proof. The concrete instantiations needs double the number of compression function calls than standard instantiations of WOTS^+.

CHAPTER 6

Merkle Signatures Schemes with Implausibility of Backdoors

Chapter 5 shows how to get few-time signature schemes with implausibility of backdoors. These schemes are restricted in that they only allow a small number (only one in the case of one-time signatures) of messages to be signed under the same verification key. They might be useful on their own in specific circumstances but signature schemes are often required to be able to sign many messages under the same verification key.

Merkle trees rely on collision resistance or second-preimage resistance to connect nodes in the tree. So far we are only able to get preimage resistance with implausibility of backdoors in the standard model. It is possible to get functions with second-preimage resistance out of functions with preimage resistance in the standard model and constructions exist under the name of Universal One-Way Hash Functions (UOWHF) [Rom90; KK05; Hai+10; Yu+15], however they are to inefficient for our purpose.

The following sections review SPR-MSS [Dah+08], a concrete Merkle tree construction also used in XMSS, and explain how results from previous sections can be used to instantiate SPR-MSS and XMSS under different assumptions with implausibility of backdoors.

6.1 Merkle Signature Schemes: SPR-MSS

SPR-MSS describes a Merkle tree that only uses second-preimage resistance instead of full collision resistance. Relying on second-preimage resistance allows more efficient constructions as parameter choices do not need to consider birthday attacks that apply to collision resistance. The practical Merkle signature scheme XMSS uses almost the same tree construction than SPR-MSS. The security of SPR-MSS applies also to XMSS.

Let $\mathsf{H} : \{0,1\}^{2n} \to \{0,1\}^n$ be a preimage resistant hash function. Let Σ be a few-time signature scheme with signature and verification key each consist of 2^l bit strings of

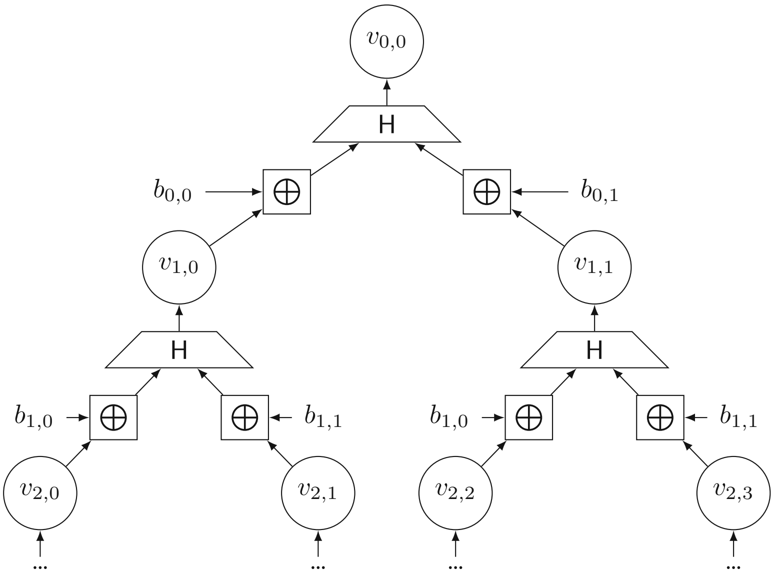

Figure 6.1: Tree structure of SPR-MSS

length n. SPR-MSS is parameterized by 2^h, the number signatures generated under one verification key. The SPR-MSS tree is then of height $h + l$. Figure 6.1 shows three layers of a SPR-MSS tree.

Key Generation: First 2^h few-time signatures keys $(\mathsf{vk}_j, \mathsf{sk}_j)$ for Σ are generated. Random bitmasks $b_{i,0}, b_{i,1} \in \{0,1\}^n$ for $i = 0, \ldots, h + l - 1$ for every level of the tree are sampled. The n-bit string of the verification keys are the leaves of the tree $v_{h+l,0}, \ldots, v_{h+l,2^{h+l}-1}$. Every node $v_{i,j}$ at height i above the leaves is computed as:

$$v_{i,j} = \mathsf{H}(v_{i+1,2j} \oplus b_{i,0} \| v_{i+1,2j+1} \oplus b_{i,1})$$

The signature key consists of the 2^h signature keys of Σ (each consisting of 2^l n-bit strings). The verification key consists the bitmasks $b_{i,0}, b_{i,1}$ for each level and the root of the tree $v_{0,0}$. If the number of elements in the verification keys of Σ are not a power of 2, the tree can simply be left unbalanced.

Signature Generation: Leaf signatures are implicitly indexed by $s \in \{0, \ldots, 2^h - 1\}$. The signature generation algorithm takes the index of the current signature s as well as the message m and signature key sk. First a few-time signature σ_s is generated by signing the message with the sth key sk_s of Σ. The sth verification key vk_s is also added to the signature. The authentication path a_s consists of the direct siblings of elements in the path from the sth signature up to the root of the tree. The resulting signature of SPR-MSS is $(s, \sigma_s, \mathsf{vk}_s, a_s)$

Signature Verification: The signature verification checks whether the few-time signature σ_s is valid for m under vk_s and whether the path up to the root can be computed with the given authentication path a_s.

SPR-MSS only needs one-time signatures but the reduction also works for few-time signatures with more than one use.

6.2 Pseudorandom Key Generation

XMSS uses almost the same tree structure as SPR-MSS. In order to decrease the size of the signature key XMSS uses pseudorandomness via a PRG with large outputs to generate the key. If the signature key is sampled uniformly random every sample needs to be stored. Computing the signature key from a uniform random seed via a PRG minimizes storage requirements to the seed only. The PRG in XMSS $\mathsf{G} : \{0,1\}^{kl} \to \{0,1\}^{n \cdot ol}$ is constructed by applying a PRF $\mathsf{F} : \{0,1\}^{kl} \times \{0,1\}^{il} \to \{0,1\}^{ol}$ to as many values as needed from a counter.

$$\mathsf{G}(s) = \mathsf{F}_s(0) \| \dots \| \mathsf{F}_s(n-1)$$

Using a PRF instead of a plain PRG allows XMSS to also compute values on demand rather than all at once during signature generation and verification.

If F is a secure PRF G is a secure PRG [BDH11]. Fischlin, Janson and Mazaheri showed a construction of a PRF with implausibility of backdoors via the HMAC construction [FJM18]. They first showed that it is implausible to include a backdoor in a weak PRF, i.e., a function family that is indistinguishable from a random function on random inputs instead of adversarially chosen ones. Unfortunately the PRG constructions does not work for a weak PRF. Let F be a secure PRF with super logarithmic input space and $c \in \{0,1\}^{ol}$ any constant in the output space. The following construction of F' is still a weak PRF.

$$\mathsf{F}'_{\mathsf{k}}(x) = \begin{cases} c & \text{for } x = 0 \\ \mathsf{F}_{\mathsf{k}}(x) & \text{else} \end{cases}$$

Instantiating the PRG G with F' however is not a secure PRG. An adversary can check whether the first output is c and thus distinguish outputs of G from uniform random strings. Thus a weak PRF is not enough for the PRG construction used in XMSS. In Chapter 5 we showed that backdoors for a PRG are implausible. As XMSS does need a PRG with large enough output, if such a secure PRG exists, it can be used directly for our purposes.

6.3 Immunizing Merkle Signature Schemes

In Chapter 5 we showed that for the concatenation combiner of two backdoored random oracles second-preimage resistance holds. The concatenation combiner can thus be used to instantiate SPR-MSS or XMSS. Like in the WOTS$^+$ example, this instantiation doubles the number of compression function calls used, because the concatenation combiner uses two different compression functions for a single message block.

Chapter 4 shows that it is implausible to backdoor second-preimage resistance of hash functions if reusability of the backdoor is considered. In this model, standard hash functions can be used to instantiate SPR-MSS or XMSS. Note that because the hash function applications have fixed size input and output, in principle only a compression function instead of a full blown hash function with domain extension is needed.

It is also conceivable to reduce the EUF-CMA security of Merkle signature schemes to preimage resistance only instead of second-preimage resistance. If this is the case, it is possible to immunize those Merkle signature schemes by a stronger implausibility result like we did in Chapter 5 for few time signatures. Our conjecture that this is possible is rooted in the fact that the preimage challenge can be embedded in Merkle trees in the same way the second-preimage challenge is in reductions for XMSS and SPHINCS variants. The downside of this approach is that signature queries that contain few time signatures contained in the subtree under the embedded preimage challenge cannot be answered. It might be possible to mitigate this by increasing the size of possible signatures by using a hyper tree, i.e., a Merkle tree with leaves signing multiple Merkle trees below, and then randomizing the choice of leaf signature in the signing process. Neither the hyper tree approach nor the randomization idea are new. At least $\mathsf{XMSS}^{\mathsf{MT}}$, XMSS-T, SPHINCS and $\mathsf{SPHINCS}^{+}$ use hyper trees and SPHINCS and $\mathsf{SPHINCS}^{+}$ use the randomization idea to get stateless hash-based signatures schemes. We did not manage to prove the conjecture due to constraints in this work but in the following give at least a preimage resistant function with implausibility of backdoors that is usable in the above context.

The preimage-resistant function needed in Merkle trees needs to be compressing. The construction from a PRG in Chapter 5, on the other hand, needs to be expanding to be preimage resistant. This gap can be bridged by a simple domain extension that retains preimage resistance but not second-preimage resistance for example. Let $\mathsf{H} : \{0,1\}^n \to \{0,1\}^{n+1}$ is a preimage resistant hash function, $\mathsf{H}'(x_1 \| x_2) = \mathsf{H}(x_1 \oplus x_2)$ is preimage resistant and $\mathsf{H}' : \{0,1\}^{2n} \to \{0,1\}^{n+1}$. It is also possible to truncate a bit from both sides of the input before computing the $\oplus$ and H operations to get $\mathsf{H}'' : \{0,1\}^{2n+2} \to \{0,1\}^{n+1}$. The output size of H'' is now half the input size and it can be used for the tree constructions. If any adversary against H' can compute a preimage $x_1' \| x_2'$ of $\mathsf{H}'(x_1 \| x_2)$, an adversary against H can use this preimage to compute a preimage under H by doing input processing itself. The same holds true for concatenation of the input.

CHAPTER 7

Signing Long Messages with Implausibility of Backdoors

The signature schemes constructed before are only able to sign small and fixed size messages. It is possible to generally transform small message signatures into signatures for much longer or even arbitrary sized messages. The tool to do that are either collision resistant hash functions or target-collision resistant hash functions.

We use two generic methods, via target-collision resistance and collision resistance, to get signatures for long messages from signatures for short messages. Target-collision resistance means it is hard for any adversary to find a collision for a target message. Functions that are designed for target-collision resistance are often called UOWHF.

7.1 Immunizing Target-Collision Resistance

Chapter 5 describes preimage resistant functions with implausibility of backdoors. UOWHFs can be constructed from any preimage resistant function [Rom90; KK05] and can be composed with a signature scheme for short messages to get a signature scheme for long messages [NY89]. Constructions of UOWHFs from general preimage resistant functions are not as efficient as practical constructions of collision resistant hash functions. Fortunately, if $\mathsf{H}(x)$ has second-preimage resistance then the function $\mathsf{H}'_{\mathsf{k}}(x) = \mathsf{H}(\mathsf{k} \oplus x)$ has target-collision resistance. The claim is easy to verify. The reduction chooses k after receiving the target messages and dependent on it. A proof can be found for example in the unfinished book of Boneh and Shoup [BS21, p. 8.11.3].

Starting from second-preimage resistant compression functions thus allows more efficient constructions. In Chapter 6 we showed that the concatenation combiner of two backdoored random oracles $C^{\mathsf{H}_1,\mathsf{H}_2}$ has second-preimage resistance, thus $C^{\mathsf{H}_1,\mathsf{H}_2}(\mathsf{k} \oplus x)$ can be used. Another way to get second-preimage resistance with implausibility of backdoors directly is to model for reusable backdoors. Chapter 4 shows that it is implausible to include a reusable backdoor in any second-preimage resistant function, thus practical compression functions assumed to be second-preimage resistant can be used.

Both schemes are only target-collision resistant for fixed length (and short) inputs. The Shoup domain extensions for target-collision resistance [Sho00] can be used to transform these into target-collision resistant hash functions for much larger inputs. The extensions grows the key size logarithmically with the size of the message to be hashed. Unfortunately keeping the key size from growing at least logarithmically seems to be impossible with target-collision resistance domain extensions [Mir01].

In the target-collision resistance game, the choice of a first message needs to be independent of the hash key used to hash this message. When composing signature schemes with target-collision resistant hash functions this translates to the need to include the hash key under the signature. Signing $(\mathsf{k}, \mathsf{H}_{\mathsf{k}}(m))$ instead of just $\mathsf{H}_{\mathsf{k}}(m)$ increases the size of the input message to the signature algorithm. With a logarithmically growing key size, this may be a problem for some signatures schemes. Cascading multiple invocations of the target-collision resistant hash functions with different keys is an option to circumvent the problem [BR97]. The first hash function compresses the input into a hash, the following hash function calls take the message hash and the hash key as an input, intuitively compressing the hash key. Every hash keys needs to be included in the signature, but only the last one actually needs to be signed.

7.2 Immunizing Collision-Resistance

If we model reusable backdoors for collision resistant hash functions, we are able to show implausibility of backdoors for collision resistant hash functions as well. Practical collision resistant hash functions can then be used with the classic hash-and-sign paradigm. In combination with the signatures schemes from the previous sections this amounts to straight forward hash-based signatures for long messages.

Bauer, Farshim and Mazaheri conjecture that the concatenation, cascade and xor combiner of backdoored random oracles also retain collision resistance [BFM18]. They give a reduction to a communication complexity theoretic problem which they assume is hard. Unfortunately there are no known results on lower bounds on this problem.

CHAPTER 8

Backdooring Hash-based Signatures

Including backdoors into hash-based signatures schemes is possible as can be seen in the following examples. A backdoored preimage-resistant hash function allows an adversary to use the backdoor to forge Lamport-Diffie signatures [Lam79]. Turning our attention away from hash-based signatures, we also describe an easy way to include a backdoor in any verification algorithm that allows an adversary to forge a signature for any signature key.

We first describe how to forge hash-based signatures by using an adversary that is able to find preimages for the used backdoored hash function. The Lamport-Diffie signature scheme serves as an easy to understand example. The technique we use can also be extended to similar hash-based few-time signatures like HORS variants.

The Lamport-Diffie signature scheme works by using preimages of elements of the verification key as signatures. Let $\mathsf{H}_\mathsf{k} : \{0,1\}^{il} \to \{0,1\}^{ol}$ be a preimage resistant hash function. The Lamport-Diffie signature scheme is parameterized by n, the length of messages that can be signed.

Key Generation: The signature key consists of $2n$ random bit strings of size il

$$\mathsf{sk} = \{s_{1,0}, s_{1,1}, \ldots, s_{n,0}, s_{n,1} \mid s_{i,j} \leftarrow\!\!\$\, \{0,1\}^{il}\}$$

The verification key is generated by applying H_k to each element in the signature key.

$$\mathsf{vk} = \{\mathsf{H}_\mathsf{k}(s_{i,j}) \mid s_{i,j} \in \mathsf{sk}\}$$

Signature Generation: A signature for the message $m = m_1 \| \dots \| m_n \in \{0,1\}^n$ is generated by publishing the elements of sk that correspond to the message bits m_i.

$$\sigma_i = \begin{cases} s_{i,0} & \text{for } m_i = 0 \\ s_{i,1} & \text{for } m_i = 1 \end{cases}$$

Signature Verification: Given the signature $\sigma = \sigma_1 \| \dots \| \sigma_n$ for the message $m = m_1 \| \dots \| m_n$ and the verification key vk, the verifier can check whether the correct preimages are provided as a signature.

$$\mathsf{H}_\mathsf{k}(\sigma_i) = \mathsf{vk}_{i,0} \text{ for } m_i = 0$$
$$\mathsf{H}_\mathsf{k}(\sigma_i) = \mathsf{vk}_{i,1} \text{ for } m_i = 1$$

As different messages differ in at least one bit, the signature includes at least one different preimage. Observe that obtaining signatures for any message and its inverse message reveals the whole signature key, thus the Lamport-Diffie signature scheme is only EUF-1-CMA secure.

Let H be a backdoored hash function with an adversary that can violate preimage resistance given the backdoor key. With access to the backdoor key and this adversary, we can request the signature $\sigma = \{\sigma_1, \dots, \sigma_n\}$ of any message $m = m_1 \| \dots \| m_n$. Flip a single bit of m at position i to get $m' = m_1 \| m_{i-1} \| m_i \oplus 1 \| m_{i+1} \| \dots \| m_n$. A valid signature of m' differs in one preimage from a valid signature of m, except for σ_i all other preimages can be reused. Use the adversary with backdoor key to compute a preimage x for the verification key on position i for m'_i. A valid signature for m' is then $\sigma' = \{\sigma_1, \dots, \sigma_{i-1}, x, \sigma_{i+1}, \dots \sigma_n\}$.

This attack strategy can be extended to HORS variants, because the relation between signature key, verification key and signature is the same. The verification key consists of the hashed signature key and signatures are elements of the signature key. The difference is in how the elements of the signature key that get published are chosen. In principle, the whole verification key can be inverted by an adversary with a backdoor and thus arbitrary messages can be signed. Requesting valid signatures beforehand alleviates us from having to compute every preimage via the adversary.

The attack described above directly applies to $\mathsf{SPR\text{-}MSS}$ and other Merkle signature schemes, if they use vulnerable few-time signature schemes as their leaf nodes. In order to forge a signature for the message m', the signature for the message m is requested. Another signature is forged under the few-time signature scheme as above. The authentication path through the Merkle tree is simply reused from the signature for m.

It is also possible to include a backdoor directly into the verification algorithm of any signature scheme. Given a EUF-CMA secure signature scheme Σ with algorithms $\Sigma = \langle \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf} \rangle$ with signatures in the space S. We create a backdoored signature scheme Σ' from Σ. The backdoor key $\mathsf{bk} \in S$ is defined as a random element in S. The backdoor in the verification algorithm Vf' is that Vf is used for all inputs except for the input bk, for which Vf' just outputs 1. Note that bk is chosen during generation of and included in the description of Σ' and is thus independent of the concrete key.

$$\mathsf{Vf}'(\mathsf{vk}, m, \sigma) = \begin{cases} 1 & \text{for } \sigma = \mathsf{bk} \\ \mathsf{Vf}(\mathsf{vk}, m, \sigma) & \text{else} \end{cases}$$

Excluding analysis of the algorithm design or implementation for a moment, the modified signature scheme is still secure, because it is hard for any adversary to find the exact input that is a valid signature for all messages and keys.

Theorem 5. *Let* $\Sigma = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$ *be a* EUF-CMA *secure signature scheme with signature space* S*. The signature scheme* $\Sigma' = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf}')$ *is an* EUF-CMA *secure signature scheme with advantage.*

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma',\mathcal{A}}(\lambda) \leq \frac{1}{|S|} + \mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma,\mathcal{B}}(\lambda)$$

Proof. We proof EUF-CMA security of Σ' by reducing to EUF-CMA of Σ. Assume an adversary $\mathcal{A}$ that violates the EUF-CMA security of Σ' We use $\mathcal{A}$ to build an adversary $\mathcal{B}$ that forges a signature for Σ.

$\mathcal{B}$ can simulate the EUF-CMA game to $\mathcal{A}$ by answering signature queries via its own signature oracle. When $\mathcal{B}$ returns a forgery (m^*, σ^*), $\mathcal{A}$ checks whether $\sigma^* = \mathsf{bk}$ and aborts if this is the case. Otherwise $\mathcal{A}$ returns (m^*, σ^*) as its own forgery. As bk is a random element of S and $\mathcal{A}$ has no information about it. The probability that $\mathcal{A}$ chooses bk as its σ^* is at most $\frac{1}{|S|}$ in which case $\mathcal{B}$ simply terminates. □

Including a signature that is valid for every message and verification key might be easy to detect by analysing the design or implementation of the signature scheme, at least without obfuscation of the design, Nonetheless, this construction shows that it is possible in principle to include a meaningful backdoor in signature schemes, warranting formal treatment of immunization techniques.

CHAPTER 9

Conclusion

We give evidence that two standard notions of hash functions, namely second-preimage resistance and collision resistance, are immune to backdoors in the design of the hash function, as long as the designer cannot allow the backdoor to be revealed by successful attacks. The last assumption aligns with the concept of “Nobody but us” (NOBUS) [Buc17], desirable by state actors or highly sophisticated adversaries that may invest considerable resources into standardizing a backdoored hash function.

Equipped with these results, we construct full hash-based signatures, from few-time signatures up to Merkle signature schemes for long messages. Unfortunately we do not reach the same level of efficiency that recent proposals of hash-based signature schemes provide. Nonetheless, we show that immunization is possible and we can reuse many techniques developed in the scope of hash-based signatures. We improve upon results on combiners of backdoored random oracles, by showing that at least the concatenation combiner is second-preimage resistant. Under this idealized model, we are able to immunize additional components of recent hash-based signature schemes. The result are more efficient schemes that are still immune to backdoors.

We leave open the question of immunizing full hash-based signature schemes without the need to not reveal the backdoor key or idealizing the underlying hash functions. Our conjecture is that a scheme similar to SPHINCS might allow this kind of immunization and sketch the solution.

List of Figures

Acronyms

ASA Algorithm Substitution Attack. 5, 6

PRF Pseudorandom Function. 2, 6, 13, 19, 22, 25, 27, 37

PRG Pseudorandom Generator. vii, ix, 2, 3, 5, 6, 22, 25–29, 37, 38

TLS Transport Layer Security. vii, ix

UOWHF Universal One-Way Hash Functions. 35, 39

Bibliography

[Alb+14] Ange Albertini, Jean-Philippe Aumasson, Maria Eichlseder, Florian Mendel, and Martin Schläffer. „Malicious Hashing: Eve's Variant of SHA-1". In: *SAC 2014: 21st Annual International Workshop on Selected Areas in Cryptography*. Ed. by Antoine Joux and Amr M. Youssef. Vol. 8781. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2014, pp. 1–19.

[AP19] Marcel Armour and Bertram Poettering. „Substitution Attacks against Message Authentication". In: *IACR Transactions on Symmetric Cryptology* 2019.3 (2019), pp. 152–168. ISSN: 2519-173X.

[Aum11] Jean-Philippe Aumasson. *Eve's SHA3 candidate: malicious hashing*. 2011. URL: https://www.aumasson.jp/data/papers/Aum11a.pdf (visited on 2021-07-26).

[AY15] Riham AlTawy and Amr M Youssef. „Watch your Constants: Malicious Streebog". In: *IET Information Security* 9.6 (2015), pp. 328–333.

[BDH11] Johannes A. Buchmann, Erik Dahmen, and Andreas Hülsing. „XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions". In: *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*. Ed. by Bo-Yin Yang. Springer, Heidelberg, 2011, pp. 117–129.

[Ber08] Daniel J. Bernstein. „The Salsa20 Family of Stream Ciphers". en. In: *New Stream Cipher Designs: The eSTREAM Finalists*. Ed. by Matthew Robshaw and Olivier Billet. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 84–97.

[Ber+15] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. „SPHINCS: Practical Stateless Hash-Based Signatures". In: *Advances in Cryptology – EUROCRYPT 2015, Part I*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. Lecture Notes in Computer Science. Springer, Heidelberg, Apr. 2015, pp. 368–397.

[Ber+19] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. „The SPHINCS$^+$ Signature Framework". In: *ACM CCS 2019: 26th Conference on Computer and Communications Security*. Ed. by Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz. ACM Press, Nov. 2019, pp. 2129–2146.

[BFM18] Balthazar Bauer, Pooya Farshim, and Sogol Mazaheri. „Combiners for Backdoored Random Oracles". In: *Advances in Cryptology – CRYPTO 2018, Part II*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10992. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2018, pp. 272–302.

[BH15] Mihir Bellare and Viet Tung Hoang. „Resisting Randomness Subversion: Fast Deterministic and Hedged Public-Key Encryption in the Standard Model". In: *Advances in Cryptology – EUROCRYPT 2015, Part II*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. Lecture Notes in Computer Science. Springer, Heidelberg, Apr. 2015, pp. 627–656.

[BJK15] Mihir Bellare, Joseph Jaeger, and Daniel Kane. „Mass-surveillance without the State: Strongly Undetectable Algorithm-Substitution Attacks". In: *ACM CCS 2015: 22nd Conference on Computer and Communications Security*. Ed. by Indrajit Ray, Ninghui Li, and Christopher Kruegel. ACM Press, Oct. 2015, pp. 1431–1440.

[BLN15] Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen. *Dual EC: A Standardized Back Door*. Cryptology ePrint Archive, Report 2015/767. 2015.

[Bos+21] Joppe W. Bos, Andreas Hülsing, Joost Renes, and Christine van Vredendaal. „Rapidly Verifiable XMSS Signatures". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021.1 (2021), pp. 137–168. ISSN: 2569-2925.

[BPR14] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. „Security of Symmetric Encryption against Mass Surveillance". In: *Advances in Cryptology – CRYPTO 2014, Part I*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2014, pp. 1–19.

[BR93] Mihir Bellare and Phillip Rogaway. „Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". In: *ACM CCS 93: 1st Conference on Computer and Communications Security*. Ed. by Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby. ACM Press, Nov. 1993, pp. 62–73.

[BR97] Mihir Bellare and Phillip Rogaway. „Collision-Resistant Hashing: Towards Making UOWHFs Practical". In: *Advances in Cryptology – CRYPTO'97*. Ed. by Burton S. Kaliski Jr. Vol. 1294. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1997, pp. 470–484.

[BS21] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. Vol. 0.5. 2021, p. 900. URL: `https://toc.cryptobook.us/` (visited on 2021-07-26).

[Buc+11a] Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hülsing, and Markus Rückert. *On the Security of the Winternitz One-Time Signature Scheme*. Cryptology ePrint Archive, Report 2011/191. 2011.

[Buc+11b] Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hülsing, and Markus Rückert. „On the Security of the Winternitz One-Time Signature Scheme“. In: *AFRICACRYPT 11: 4th International Conference on Cryptology in Africa*. Ed. by Abderrahmane Nitaj and David Pointcheval. Vol. 6737. Lecture Notes in Computer Science. Springer, Heidelberg, July 2011, pp. 363–378.

[Buc17] Ben Buchmann. „Nobody but us“. In: *The rise and fall of the golden age of signals intelligence. A Hoover Institute Essay. Aegis Series Paper* 1708 (2017).

[Cam+20] Fabio Campos, Tim Kohlstadt, Steffen Reith, and Marc Stöttinger. „LMS vs XMSS: Comparison of Stateful Hash-Based Signature Schemes on ARM Cortex-M4“. In: *AFRICACRYPT 20: 12th International Conference on Cryptology in Africa*. Ed. by Abderrahmane Nitaj and Amr M. Youssef. Vol. 12174. Lecture Notes in Computer Science. Springer, Heidelberg, July 2020, pp. 258–277.

[Che+14] Stephen Checkoway, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham, and Matthew Fredrikson. „On the Practical Exploitability of Dual EC in TLS Implementations“. In: *USENIX Security 2014: 23rd USENIX Security Symposium*. Ed. by Kevin Fu and Jaeyeon Jung. USENIX Association, Aug. 2014, pp. 319–335.

[Che+16] Stephen Checkoway, Jacob Maskiewicz, Christina Garman, Joshua Fried, Shaanan Cohney, Matthew Green, Nadia Heninger, Ralf-Philipp Weinmann, Eric Rescorla, and Hovav Shacham. „A Systematic Analysis of the Juniper Dual EC Incident“. In: *ACM CCS 2016: 23rd Conference on Computer and Communications Security*. Ed. by Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi. ACM Press, Oct. 2016, pp. 468–479.

[Cvea] *CVE-2008-0166*. 2008. URL: `https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2008-0166` (visited on 2021-07-26).

[Cveb] *CVE-2014-0160*. 2014. URL: `https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2008-0160` (visited on 2021-07-26).

[Dah+08] Erik Dahmen, Katsuyuki Okeya, Tsuyoshi Takagi, and Camille Vuillaume. „Digital Signatures Out of Second-Preimage Resistant Hash Functions“. In: *Post-quantum cryptography, second international workshop, PQCRYPTO 2008*. Ed. by Johannes Buchmann and Jintai Ding. Springer, Heidelberg, Oct. 2008, pp. 109–123.

[Dam90] Ivan Damgård. „A Design Principle for Hash Functions“. In: *Advances in Cryptology – CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1990, pp. 416–427.

[Dod+15] Yevgeniy Dodis, Chaya Ganesh, Alexander Golovnev, Ari Juels, and Thomas Ristenpart. „A Formal Treatment of Backdoored Pseudorandom Generators“. In: *Advances in Cryptology – EUROCRYPT 2015, Part I*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. Lecture Notes in Computer Science. Springer, Heidelberg, Apr. 2015, pp. 101–126.

[Dod+20] Yevgeniy Dodis, Pooya Farshim, Sogol Mazaheri, and Stefano Tessaro. „Towards Defeating Backdoored Random Oracles: Indifferentiability with Bounded Adaptivity“. In: *TCC 2020: 18th Theory of Cryptography Conference, Part III*. Ed. by Rafael Pass and Krzysztof Pietrzak. Vol. 12552. Lecture Notes in Computer Science. Springer, Heidelberg, Nov. 2020, pp. 241–273.

[DSS05] C. Dods, Nigel P. Smart, and Martijn Stam. „Hash Based Digital Signature Schemes“. In: *10th IMA International Conference on Cryptography and Coding*. Ed. by Nigel P. Smart. Vol. 3796. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2005, pp. 96–115.

[Dwo15] Morris Dworkin. „SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions“. en. In: *Federal Inf. Process. Stds. (NIST FIPS)* (2015).

[FJM18] Marc Fischlin, Christian Janson, and Sogol Mazaheri. „Backdoored Hash Functions: Immunizing HMAC and HKDF“. In: *CSF 2018: IEEE 31st Computer Security Foundations Symposium*. Ed. by Steve Chong and Stephanie Delaune. IEEE Computer Society Press, 2018, pp. 105–118.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. „How to Construct Random Functions“. In: *Journal of the ACM* 33.4 (Oct. 1986), pp. 792–807.

[Hai+10] Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. „Universal One-Way Hash Functions via Inaccessible Entropy“. In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. Lecture Notes in Computer Science. Springer, Heidelberg, 2010, pp. 616–637.

[Hås+99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. „A Pseudorandom Generator from any One-way Function“. In: *SIAM Journal on Computing* 28.4 (1999), pp. 1364–1396.

[HRB13] Andreas Hülsing, Lea Rausch, and Johannes Buchmann. „Optimal Parameters for XMSS^{MT}". In: *Security Engineering and Intelligence Informatics*. Ed. by Alfredo Cuzzocrea, Christian Kittl, Dimitris E. Simos, Edgar Weippl, and Lida Xu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 194–208.

[HRS16] Andreas Hülsing, Joost Rijneveld, and Fang Song. „Mitigating Multi-target Attacks in Hash-Based Signatures". In: *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I*. Ed. by Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang. Vol. 9614. Lecture Notes in Computer Science. Springer, Heidelberg, Mar. 2016, pp. 387–416.

[Hül13] Andreas Hülsing. „W-OTS+ - Shorter Signatures for Hash-Based Signature Schemes". In: *AFRICACRYPT 13: 6th International Conference on Cryptology in Africa*. Ed. by Amr Youssef, Abderrahmane Nitaj, and Aboul Ella Hassanien. Vol. 7918. Lecture Notes in Computer Science. Springer, Heidelberg, June 2013, pp. 173–188.

[Hül+18] Andreas Hülsing, Denis Butin, Stefan-Lukas Gazdag, Joost Rijneveld, and Aziz Mohaisen. *XMSS: eXtended Merkle Signature Scheme*. RFC 8391. RFC Editor, 2018. URL: `https://www.rfc-editor.org/rfc/rfc8391.txt`.

[IR89] Russell Impagliazzo and Steven Rudich. „Limits on the Provable Consequences of One-Way Permutations". In: *21st Annual ACM Symposium on Theory of Computing*. ACM Press, May 1989, pp. 44–61.

[KBC97] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104. RFC Editor, 1997. URL: `http://www.rfc-editor.org/rfc/rfc2104.txt`.

[KE10] Hugo Krawczyk and Pasi Eronen. *HMAC-based Extract-and-Expand Key Derivation Function (HKDF)*. RFC 5869. RFC Editor, 2010. URL: `http://www.rfc-editor.org/rfc/rfc5869.txt`.

[KK05] Jonathan Katz and Chiu-Yuen Koo. *On Constructing Universal One-Way Hash Functions from Arbitrary One-Way Functions*. Cryptology ePrint Archive, Report 2005/328. 2005.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. 2nd. Chapman & Hall/CRC, 2014. ISBN: 1466570261.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. „Signature Schemes with Bounded Leakage Resilience". In: *Advances in Cryptology – ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. Lecture Notes in Computer Science. Springer, Heidelberg, Dec. 2009, pp. 703–720.

[Lam79] Leslie Lamport. *Constructing Digital Signatures from a One-way Function*. Technical Report SRI-CSL-98. SRI International Computer Science Laboratory, Oct. 1979.

[LM17] Philip Lafrance and Alfred Menezes. *On the security of the WOTS-PRF signature scheme*. Cryptology ePrint Archive, Report 2017/938. 2017.

[Maz20] Sogol Mazaheri. „Cryptographic Primitives that Resist Backdooring and Subversion“. PhD thesis. Darmstadt: Technische Universität, 2020. URL: `http://tuprints.ulb.tu-darmstadt.de/14550/`.

[Mer88] Ralph C. Merkle. „A Digital Signature Based on a Conventional Encryption Function“. In: *Advances in Cryptology – CRYPTO'87*. Ed. by Carl Pomerance. Vol. 293. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1988, pp. 369–378.

[Mer90a] Ralph C. Merkle. „A Certified Digital Signature“. In: *Advances in Cryptology – CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1990, pp. 218–238.

[Mer90b] Ralph C. Merkle. „One Way Hash Functions and DES“. In: *Advances in Cryptology – CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1990, pp. 428–446.

[Mir01] Ilya Mironov. „Hash Functions: From Merkle-Damgård to Shoup“. In: *Advances in Cryptology – EUROCRYPT 2001*. Ed. by Birgit Pfitzmann. Vol. 2045. Lecture Notes in Computer Science. Springer, Heidelberg, May 2001, pp. 166–181.

[Mor15] Pawel Morawiecki. *Malicious Keccak*. Cryptology ePrint Archive, Report 2015/1085. 2015.

[NIS] NIST National Institute of Standards and Technology. *NIST Post-Quantum Cryptography Project*. URL: `https://csrc.nist.gov/Projects/post-quantum-cryptography` (visited on 2021-07-26).

[NY89] Moni Naor and Moti Yung. „Universal One-Way Hash Functions and their Cryptographic Applications“. In: *21st Annual ACM Symposium on Theory of Computing*. ACM Press, May 1989, pp. 33–43.

[PA21] Raluca Posteuca and Tomer Ashur. *How to Backdoor a Cipher*. Cryptology ePrint Archive, Report 2021/442. 2021.

[PW20] Thomas Peyrin and Haoyang Wang. „The MALICIOUS Framework: Embedding Backdoors into Tweakable Block Ciphers“. In: *Advances in Cryptology – CRYPTO 2020, Part III*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2020, pp. 249–278.

[PWX04] Josef Pieprzyk, Huaxiong Wang, and Chaoping Xing. „Multiple-Time Signature Schemes against Adaptive Chosen Message Attacks“. In: *SAC 2003: 10th Annual International Workshop on Selected Areas in Cryptography*. Ed. by Mitsuru Matsui and Robert J. Zuccherato. Vol. 3006. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2004, pp. 88–100.

[Res18] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. RFC Editor, 2018. URL: `https://www.rfc-editor.org/rfc/rfc8446.txt`.

[Rom90] John Rompel. „One-Way Functions are Necessary and Sufficient for Secure Signatures“. In: *22nd Annual ACM Symposium on Theory of Computing*. ACM Press, May 1990, pp. 387–394.

[RP97] Vincent Rijmen and Bart Preneel. „A Family of Trapdoor Ciphers“. In: *Fast Software Encryption – FSE'97*. Ed. by Eli Biham. Vol. 1267. Lecture Notes in Computer Science. Springer, Heidelberg, Jan. 1997, pp. 139–148.

[RR02] Leonid Reyzin and Natan Reyzin. „Better than BiBa: Short One-Time Signatures with Fast Signing and Verifying“. In: *ACISP 02: 7th Australasian Conference on Information Security and Privacy*. Ed. by Lynn Margaret Batten and Jennifer Seberry. Vol. 2384. Lecture Notes in Computer Science. Springer, Heidelberg, July 2002, pp. 144–153.

[RS04] Phillip Rogaway and Thomas Shrimpton. „Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance“. In: *Fast Software Encryption – FSE 2004*. Ed. by Bimal K. Roy and Willi Meier. Vol. 3017. Lecture Notes in Computer Science. Springer, Heidelberg, Feb. 2004, pp. 371–388.

[Sho00] Victor Shoup. „A Composition Theorem for Universal One-Way Hash Functions“. In: *Advances in Cryptology – EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. Lecture Notes in Computer Science. Springer, Heidelberg, May 2000, pp. 445–452.

[Wan+19] Wen Wang, Bernhard Jungk, Julian Wälde, Shuwen Deng, Naina Gupta, Jakub Szefer, and Ruben Niederhagen. „XMSS and Embedded Systems“. In: *SAC 2019: 26th Annual International Workshop on Selected Areas in Cryptography*. Ed. by Kenneth G. Paterson and Douglas Stebila. Vol. 11959. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2019, pp. 523–550.

[Yu+15] Yu Yu, Dawu Gu, Xiangxue Li, and Jian Weng. „(Almost) Optimal Constructions of UOWHFs from 1-to-1, Regular One-Way Functions and Beyond“. In: *Advances in Cryptology – CRYPTO 2015, Part II*. Ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9216. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 2015, pp. 209–229.

[YY96] Adam Young and Moti Yung. „The Dark Side of "Black-Box" Cryptography, or: Should We Trust Capstone?“ In: *Advances in Cryptology – CRYPTO'96*. Ed. by Neal Koblitz. Vol. 1109. Lecture Notes in Computer Science. Springer, Heidelberg, Aug. 1996, pp. 89–103.

[YY97] Adam Young and Moti Yung. „Kleptography: Using Cryptography Against Cryptography“. In: *Advances in Cryptology – EUROCRYPT'97*. Ed. by Walter Fumy. Vol. 1233. Lecture Notes in Computer Science. Springer, Heidelberg, May 1997, pp. 62–74.