ROBERT ANNESSI

SECURING GROUP COMMUNICATION IN CRITICAL INFRASTRUCTURES

DISSERTATION

ADVISOR: TANJA ZSEBY (TU WIEN)
ASSISTANCE: JOACHIM FABINI (TU WIEN)
REVIEWERS: ADRIAN PERRIG (ETH ZURICH)
EDGAR WEIPPL (TU WIEN)

TECHNISCHE UNIVERSITÄT WIEN

January 2019

Contents

1 Introduction 12

1.1 Problem Statement 12
1.2 Research Questions 13
1.3 Methodology 14
1.4 Contributions 14

2 Background 17

2.1 Hash Functions 19
2.2 One-Way Chains 19
2.3 Message Authentication Codes 20
2.4 Digital Signatures 21
2.5 Group Authentication and Data Origin Authentication 23
2.6 The EdDSA Signature Scheme 24
2.7 MQ Signature Schemes 25
2.8 Clock Synchronization 26
2.9 Subliminal Communication 29
2.10 Botnets 30

3 Related Work 31

3.1 Data Origin Authentication for Group Communication 32
3.2 Secure Clock Synchronization 32
3.3 Subliminal Communication 35

4 Threat Model 37

4.1 Network Model 37
4.2 Adversary Model 38
4.3 Application Model 38
4.4 Attacks 38

5 Use Cases 40
5.1 Clock Synchronization 40
5.2 5G Networks 42
5.3 Sensor Data Collection in Smart Grids 45

6 Requirements 46
6.1 Performance 46
6.2 Security 48
6.3 Robustness 48

7 Classification of Data Origin Authentication Schemes 50
7.1 Extending Symmetric Schemes to Data Origin Authentication 51
7.2 Reducing the Cost of Conventional Signature Schemes 51
7.3 Designing Fast Authentication Schemes 52

8 Evaluation of Data Origin Authentication Schemes 56
8.1 Extending Symmetric Schemes for Data Origin Authentication 57
8.2 Reducing the Cost of Conventional Signature Schemes 58
8.3 Designing Fast Authentication Schemes 65
8.4 Summary 70

9 Assessment of Unrestricted-Time High-Speed Signing 75

10 Secure Clock Synchronization 77
10.1 Delay Attacks 79
10.2 SecureTime Protocol 98

11 Group Communication in 5G Networks 106

12 Sensor Data Collection in Smart Grids 110

13 Subliminal Channels 112
13.1 Subliminal Channels in EdDSA 113

13.2 Subliminal Communication Scenarios 114
13.3 Practical Experiments with Subliminal Communication 117
13.4 Preventing Subliminal Communication 119
13.5 Subliminal Channels in MQ-based Signature Schemes 122
13.6 ChainChannels: Subliminal Channels in Blockchains 124

14 Conclusion 134

Acronyms 136

Bibliography 139

List of Figures

2.1 Transmission system categories. 18
2.2 Goals in information security and their relation to data origin authentication. 19
2.3 Generation and use of one-way chains. 20
2.4 Message authentication using a shared key. 21
2.5 Digital signature using an asymmetric key pair. 22
2.6 The basic operation of MQ-based signatures. 25
2.7 Two-step clock synchronization. 28
2.8 Subliminal communication in digital signatures. 29

5.1 Security measures related to group communication in 5G. 44

7.1 Classification of data origin authentication schemes for group communication. 51
7.2 Merkle tree construction. 54

8.1 Timeline of secret-information asymmetry schemes. 57
8.2 Timeline of deferred signing schemes. 59
8.3 Timeline of signature propagation schemes. 60
8.4 Basic principle of signature propagation schemes. 61
8.5 Timeline of signature dispersal schemes. 63
8.6 Basic principle of signature dispersal schemes. 64
8.7 Timeline of OTS schemes. 65
8.8 Timeline of MTS schemes. 66

10.1 Delay attack on TESLA as part of a clock synchronization protocol. 78
10.2 PTP clock synchronization with asymmetric delay. 80
10.3 PTP clock synchronization where SYNC message is affected by additional delay. 81
10.4 Precision Time Protocol (PTP) traffic patterns in timing, length, and direction. 83
10.5 Experimental setup used to examine the effect of delay attacks on PTP over an untrusted network. 85
10.6 Offset to UTC during an selective SYNC message delay attack. 86
10.7 Offset to UTC during a selective DELAY_REQUEST message delay attack. 86
10.8 Offset to UTC during a incremental selective DELAY_REQUEST message delay attack. 87
10.9 Asymmetric one-way delay attack where the master→slave delay is altered. 90
10.10 Asymmetric one-way delay attack where the slave→master delay is altered. 91
10.11 One-way delays in clock synchronization. 91
10.12 Theoretical (worst case) offset uncertainty bound calculation. 93
10.13 Offset uncertainty bound calculation using one-way delay limits. 94
10.14 Measurement of signed NTP messages. 99

11.1 Short-range attack scenario on group communication services in 5G. 107
11.2 Long-range attack scenario on group communication services in 5G. 108

12.1 PMU sensor data multicasted to various applications in a Smart Grid. 110

13.1 Subliminal communication scenario: Botnet C&C using NTP broadcasts. 115
13.2 Subliminal communication scenario: Information leakage through phasor measurements. 115
13.3 Handshake in TLS version 1.3 and before. 116
13.4 Experimental setup for investigating a subliminal channel in signed broadcast NTP messages. 117
13.5 ChainChannels: Subliminal message propagation. 126
13.6 ChainChannels: Key leakage based on secret sharing 128
13.7 ChainChannels: Concealed key leakage. 129

List of Tables

1.1 Summary of research questions, methodologies, and major findings. 16

2.1 Comparison of symmetric and asymmetric signing schemes. 22
2.2 Group authentication vs. data origin authentication. 23

4.1 Summary of the threat model. 37

5.1 Clock synchronization threat analysis. 41

6.1 Evaluation criteria for data origin authentication schemes in group communication. 47

8.1 Theoretical performance evaluation of data origin authentication schemes. 72
8.2 Theoretical security evaluation of data origin authentication schemes. 73
8.3 Theoretical robustness evaluation of data origin authentication schemes. 74

9.1 Computational efficiency and communication overhead of high-speed signature schemes compared to TV-HORS. 76

10.1 Notation used for delay attacks and SecureTime. 79
10.2 Identified properties of PTP traffic. 84
10.3 Countermeasures against selective message delay attacks. 89
10.4 Measurement results: Network Time Protocol (NTP)'s clock synchronization precision. 100
10.5 Comparison of SecureTime to related work. 101

13.1 Measurement results for three subliminal communication scenarios. 118
13.2 Methods to provide subliminal-freeness. 121
13.3 Subliminal bandwidths of MQ signature schemes. 123
13.4 Values for ChainChannel's proof of concept. 130

Abstract

Communication networks have become an essential part of increasingly interconnected modern societies. Group communication is a ubiquitous concept in today's communication networks, and comprises broadcast, multicast, and anycast communication. Since group communication facilitates efficient data transmission to numerous receivers, it is more and more needed generally and specifically in critical infrastructures such as sensor data collection in Smart Grids, clock synchronization, and 5G networks. Surprisingly, no generally applicable method exists as yet to secure group communication from adversarial attacks. For this reason, group communication is often times either not secured at all or application-specific security measures are deployed that are not generally applicable and whose security is hard to assess.

In this thesis, we tackle a fundamental challenge in securing group communication: data origin authentication. We evaluate various data origin authentication schemes that were proposed during the last twenty-five years for their suitability to secure group communication for critical infrastructures in general and suggest a new classification for data origin authentication schemes that covers developments in recent years. With the advent of novel high-speed signature schemes, we furthermore suggest a new class of data origin authentication schemes: unrestricted-time high-speed signing. In this way, we revise the common assumption that signing every packet individually is computationally unfeasible. To validate the unrestricted-time high-speed signing class suggested in this thesis, we evaluate it for a set of applications in critical infrastructures: sensor data collection in Smart Grids, group communication in 5G networks, and clock synchronization. For clock synchronization we additionally propose a novel set of security measures against a wealth of attacks including delay attacks and discover a fundamental limitation in clock synchronization protocols: they can either be precise or secure.

An additional challenge may become prevalent when data origin authentication schemes are used on a large scale or in high-speed environments: subliminal channels in signatures. We analyze several high-speed signature schemes for their susceptibility to subliminal channels and find all of them to be susceptible. As a proof of concept, we introduce a method that exploits such subliminal channel for private botnet command and control communication over public blockchains. Given the results on data origin authentication, subliminal channels, and clock synchronization, we are confident that this thesis contributes to the foundation of secure group communication in critical infrastructures.

Note of adoption from previous publications

This thesis has been published in part in:

Robert Annessi, Tanja Zseby, and Joachim Fabini. "A new Direction for Research on Data Origin Authentication in Group Communication". In: *International Conference on Cryptology and Network Security (CANS)*. Springer, 2017. DOI: 10.1007/978-3-030-02641-7_26. Chapters 2, 3, 6, 7, 8, 9. Referred to as "[I]". Adapted with permission from Springer Nature Customer Service Centre GmbH. © Springer Nature 2018.

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Chapters 2, 3, 4, 5, 6, 7, 8, 9, 10. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Chapters 2, 3, 4, 5, 6, 9, 10. Referred to as "[III]".

Robert Annessi, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252. Chapters 2, 5, 6, 7, 8, 9, 11. Referred to as "[IV]". © ACM 2018.

Davor Frkat, Robert Annessi, and Tanja Zseby. "ChainChannels: Private Botnet Communication Over Public Blockchains". In: *IEEE International Conference on Blockchain (Blockchain)*. 2018. Chapters 2, 3, 13. Referred to as "[V]". © IEEE 2018.

Alexander Hartl, Robert Annessi, and Tanja Zseby. "A Subliminal Channel in EdDSA: Information Leakage with High-Speed Signatures". In: *International Workshop on Managing Insider Security Threats*. MIST '17. Dallas, Texas, USA: ACM, 2017, pp. 67–78. ISBN: 978-1-4503-5177-5. DOI: 10.1145/3139923.3139925. Chapters 2, 3, 13. Referred to as "[VI]". © ACM 2017.

Alexander Hartl, Robert Annessi, and Tanja Zseby. "Subliminal Channels in High-Speed Signatures". In: *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)* 9.1 (Mar. 2018), pp. 30–53. Chapters 2, 3, 13. Referred to as "[VII]". © JoWUA 2018.

Robert Annessi, Joachim Fabini, Felix Iglesias, and Tanja Zseby. *Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization*. 2018. arXiv: 1811.08569 [cs.CR]. Chapters 2, 3, 4, 5, 10. Referred to as "[VIII]".

Attribution to coauthors of joint publications

Several parts of the joint papers were conducted in close cooperation with the coauthors. The experiments on identifying PTP messages in encrypted traffic with statistical methods (described in the joint paper [VIII] and Subsection 10.1.1) was conducted primarily by Félix Iglesias Vázquez. The work on subliminal channels in EdDSA (described in the joint papers [VI], [VII], and Sections 13.1, 13.2, 13.3, 13.4, and 13.5) was conducted in collaboration with Alexander Hartl. The practical experiments (described in Section 13.3) and the work on subliminal channels in MQ-based signature schemes (described in Section 13.5) were conducted primarily by Alexander Hartl. The work on ChainChannels (described in the joint paper [V] and Section 13.6) was conducted in collaboration with Davor Frkat. The practical implementation was conducted primarily by Davor Frkat.

(Co)authored publications that are not part of this thesis:

Robert Annessi and Martin Schmiedecker. "NavigaTor: Finding Faster Paths to Anonymity". In: *IEEE European Symposium on Security and Privacy (Euro S&P)*. 2016. DOI: 10.1109/EuroSP.2016.26

Felix Iglesias Vazquez, Robert Annessi, and Tanja Zseby. "DAT detectors: uncovering TCP/IP covert channels by descriptive analytics". In: *Security And Communication Networks* (2016). DOI: 10.1002/sec.1531

Tanja Zseby, Felix Iglesias Vazquez, Valentin Bernhardt, Davor Frkat, and Robert Annessi. "A Network Steganography Lab on Detecting TCP/IP Covert Channels". In: *IEEE Transactions on Education* PP.99 (2016), pp. 1–9. DOI: 10.1109/TE.2016.2520400

Felix Iglesias, Valentin Bernhardt, Robert Annessi, and Tanja Zseby. "Decision Tree Rule Induction for Detecting Covert Timing Channels in TCP/IP Traffic". In: *International Cross Domain Conference for Machine Learning & Knowledge Extraction (CD-MAKE)*. 2017

Felix Iglesias Vazquez, Robert Annessi, and Tanja Zseby. "Analytic Study of Features for the Detection of Covert Timing Channels in Network Traffic". In: *Journal of Cyber Security and Mobility* 6.3 (2017), pp. 225–270. DOI: 10.13052/jcsm2245-1439.632

1
Introduction

GROUP COMMUNICATION is ubiquitous in today's communication networks. It facilitates efficient data transmission to numerous receivers by minimizing data replication efforts as well as load, both at the sender and in the network. Unicast communication, on the other hand, is efficient when receivers consume different content at different times such as video on-demand but does not scale well when many receivers consume the same content at the same time such as a live video streaming. Whenever communications conform to such simultaneity constraints, group communication provides a method for transmitting data efficiently from one sender to possibly many receivers without running into scalability issues.

Due to the expansion of communications, group communication has manifold applications in today's networks from content broadcasting (e.g., TV and radio), voice and video conferencing, distribution of information such as stock market information or software updates, self-configuration and replication of distributed systems such as cache synchronization in Content Delivery Networks (CDNs), Peer to Peer (P2P) networks, Massively Multiplayer Online Games (MMOGs), Multicast Domain Name System (DNS) (mDNS), software updates, wireless sensor networks, to military defense systems. Group communication is a fundamental concept that is implemented on different abstraction layers of the conceptual Open Systems Interconnection (OSI) model: data link (Ethernet, Asynchronous Transfer Mode (ATM), or Infiniband), network (IPv4, IPv6), and application layer using overlay networks.

Surprisingly, no generally applicable method exists to secure group communication from adversarial attacks. In this thesis, we investigate the fundamental building block of secure group communication — data origin authentication — with the vision of laying the foundation of a security protocol for group communication, comparable to what the Transport Layer Security (TLS) protocol is for unicast communication today.

1.1 Problem Statement

The security of communications is of paramount importance, specifically in critical infrastructures, which have become more and more depended on communication networks. One specific example of such critical infrastructure is the future power grid, commonly referred to as *Smart Grid*, in which power, communication, and information technologies are integrated for an improved

electrical power infrastructure[1]. To provide exact control and in-time anomaly detection in Smart Grids, data must be relayed fast with low processing delay over long distances to multiple receivers. Given the nature of critical decisions based on measurements, authenticity is of crucial importance, in particular to Wide Area Monitoring, Protection, and Control Systems (WAMPACSs) which are used to enhance situational awareness in the power grid[2,3,4]. Another example for critical infrastructures that depend on communication networks are clock synchronization protocols such as the Network Time Protocol (NTP)[5] and the Precision Time Protocol (PTP)[6].

Group communication comprises various challenges, many of which stem from its unidirectional nature and dynamic group membership. Some challenges can be solved easier on higher abstraction layers, such as guaranteeing reliable delivery of packets. Other challenges tend to reoccur, such as efficient and secure authentication of senders, no matter on which abstraction layer group communication functionality is implemented. One reoccurring, fundamental challenge in group communication—the authentication of the sender—is called *data origin authentication* (sometimes still referred to as source authentication, a term considered deprecated[7]).

Cryptographic methods have been typically designed for unicast communication and applying them to group communication yields inefficient and non-adequate solutions[8,9]. Despite more than twenty-five years of research on data origin authentication for group communication, in which various data origin authentication schemes have been proposed, no sufficiently efficient and secure scheme as yet exists that could be deployed generally on a large scale. None of the proposed schemes satisfies all constraints and requirements of applications, and naming a single superior scheme seems non-trivial[10] so that data origin authentication for group communication remains challenging. For this reason, application-specific solutions were developed that may comprise sub-optimal or even insecure data origin authentication schemes. As long as there is no appropriate solution to the data origin authentication challenge, many application developers either have to make weaker security guarantees or cannot benefit from improved efficiency through group communication in the first place.

1.2 Research Questions

In this thesis, we seek to answer the following research questions:

- Which properties should data origin authentication schemes provide in order to make them generally applicable to critical infrastructure applications?
- Which state-of-the-art data origin authentication schemes provide these properties (or can be modified in such way)?
- Can high-speed signatures build a foundation for a generally applicable data origin authentication schemes that provide all properties?
- Are subliminal channels (in high-speed signatures) a significant risk and can subliminal channels be prevented?
- What measures are needed to secure clock synchronization?

[1] "IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads". In: *IEEE Std 2030-2011* (Sept. 2011), pp. 1–126. DOI: 10.1109/IEEESTD.2011.6018239.

[2] Qiyan Wang et al. "Time Valid One-Time Signature for Time-Critical Multicast Data Authentication". In: *IEEE INFOCOM*. Apr. 2009, pp. 1233–1241. DOI: 10.1109/INFCOM.2009.5062037.

[3] Tanja Zseby and Joachim Fabini. "Security Challenges for Wide Area Monitoring in Smart Grids". In: *e & i Elektrotechnik und Informationstechnik* 131.3 (May 2014), pp. 105–111. ISSN: 0932-383X, 1613-7620. DOI: 10.1007/s00502-014-0203-3.

[4] T.H. Morris, Shengyi Pan, and U. Adhikari. "Cyber security recommendations for wide area monitoring, protection, and control systems". In: *IEEE Power and Energy Society General Meeting*. July 2012, pp. 1–6. DOI: 10.1109/PESGM.2012.6345127.

[5] D. Mills et al. *Network Time Protocol Version 4: Protocol and Algorithms Specification*. RFC 5905 (Proposed Standard). RFC. Updated by RFC 7822. Fremont, CA, USA: RFC Editor, June 2010. DOI: 10.17487/RFC5905. URL: https://www.rfc-editor.org/rfc/rfc5905.txt.

[6] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems". In: *IEEE Std 1588-2008* (July 2008), pp. 1–269. DOI: 10.1109/IEEESTD.2008.4579760.

[7] R. Shirey. *Internet Security Glossary, Version 2*. RFC 4949 (Informational). RFC. Fremont, CA, USA: RFC Editor, Aug. 2007. DOI: 10.17487/RFC4949. URL: https://www.rfc-editor.org/rfc/rfc4949.txt.

[8] Y. Challal, H. Bettahar, and A. Bouabdallah. "A taxonomy of multicast data origin authentication: Issues and solutions". In: *IEEE Communications Surveys Tutorials* 6.3 (2004), pp. 34–57. ISSN: 1553-877X. DOI: 10.1109/COMST.2004.5342292.

[9] P. Judge and M. Ammar. "Security issues and solutions in multicast content distribution: a survey". In: *IEEE Network* 17.1 (Jan. 2003), pp. 30–36. ISSN: 0890-8044. DOI: 10.1109/MNET.2003.1174175.

[10] Rainer Steinwandt and Viktória I. Villányi. "A One-time Signature Using Run-length Encoding". In: *Information Processing Letters* 108.4 (Oct. 2008), pp. 179–185. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2008.05.004.

1.3 Methodology

To tackle these questions, we first define a threat model for secure group communication (Chapter 4) and then analyze different use cases for group communication: clock synchronization, sensor data collection in Smart Grids, and 5G cellular networks (Chapter 5); we also discuss group communication in blockchains, a potential building block for future applications in critical infrastructures (Section 13.6). From these use cases we derive a set of requirements for data origin authentication schemes that satisfies all analyzed applications (Chapter 6). Because of the sheer number of data origin authentication schemes that have been proposed in the last twenty-five years, we classify them first (Chapter 7) before conducting a theoretical evaluation (Chapter 8). In Chapter 9, we assess our new class of schemes—*unrestricted-time high-speed signing*—based on novel high-speed digital signature schemes and evaluate our class in Chapters 10, 11, and 12 regarding the use cases. One side effect of data origin authentication schemes are subliminal channels that may be used to clandestinely transmit information to a third party. While the problem of subliminal channels is known, it is elevated and becomes more severe in future scenarios with data origin authentication (especially when signing large amounts of streaming data) as evaluated both theoretically and practically in Chapter 13.

1.4 Contributions

This thesis has following major contributions that are summarized in Table 1.1 on page 16:

Data Origin Authentication We suggest a new classification for data origin authentication schemes and relate them to three basic research approaches we identified. We argue that the most promising research approach for securing group communication is to design fast authentication schemes. Our evaluation of data origin authentication schemes shows that each class of schemes comprises a trade-off from a specific point of view. When analyzing security solutions for applications, we find that authenticity in group communication is often done wrong: either only group authentication is provided instead of data origin authentication or unsuitable data origin authentication schemes are employed. In this thesis, we show such shortcomings in two applications we checked, group communication in 5G networks (Chapter 11) and clock synchronization (Chapter 10). Furthermore, we suggest a new class of schemes—unrestricted-time high-speed signing—that follows the approach of designing fast authentication schemes and show that it significantly elevates the solution space and provides a general solution to the authentication challenge in group communication.

Clock Synchronization We are the first to conduct a comprehensive theoretical evaluation of data origin authentication schemes regarding their suitability to secure multicast clock synchronization. Based on our suggested unrestricted-time high-speed signing class, we propose *SecureTime*, a set of measures to secure (multicast) clock synchronization (Section 10.2) against substitution and

impersonation attacks and propose an additional set of security measures to prevent replay attacks as well as a novel method to mitigate delay attacks. For delay attacks, we give upper bounds on the delays that can be introduced maliciously by an adversary. When researching clock synchronization security, we found a fundamental relation: clock synchronization protocols can either be precise or secure but not both! Based on this insight, we derive a formula to calculate the clock offset bounds that can be guaranteed for a particular system in adversarial settings. We conduct performance measurements and show that using unrestricted-time high-speed signing has low computational overhead, low impact on the precision in 1-step mode, and introduces only low communication overhead per message. In 2-step mode, SecureTime has practically even zero impact on clock synchronization's precision while the communication overhead is just moderately increased.

Subliminal Communication When using signatures for data origin authentication in group communication, the possibility to transmit subliminal information may become a severe challenge. We show subliminal channels to exist in EdDSA and Multivariate Quadratic (MQ)-based signature schemes (the latter provide post-quantum security) and show methods to exploit them including clandestine ways to exchange needed key material. We analyze the applicability of the subliminal channel in practical experiments for different scenarios. One of the scenarios are blockchains, for which we introduce *ChainChannels*, a novel method to create a hidden Command & Control (C&C) infrastructure for multicasting information to bots by transmitting subliminal information in blockchain signatures. Our method is neither restricted to C&C nor to a particular blockchain and therefore provides a general method for hidden distribution channels over blockchains that can be applied to other scenarios in which communication channels should remain hidden. We identify countermeasures that ensure subliminal-freeness but show that none of the countermeasures is generally viable in the context of network protocols so that protecting information assets from leakage remains a major challenge.

Research question	Methodology	Major findings	Chapters
Which properties should data origin authentication schemes provide in order to make them generally applicable to critical infrastructure applications?	Analysis of different use cases for group communication in critical infrastructures: clock synchronization, sensor data collection in Smart Grids, and 5G cellular networks.	Comprehensive set of properties in three categories: performance, security, and robustness.	5 and 6
Which state-of-the-art data origin authentication schemes provide these properties (or can be modified in such way)?	Theoretical evaluation of state-of-the-art data origin authentication schemes.	Not a single state-of-the-art data origin authentication scheme provides all properties required for generally applicability. We suggest a new classification for data origin authentication schemes.	7 and 8
Can high-speed signatures build a foundation for a generally applicable data origin authentication schemes that provide all properties?	Theoretical and practical evaluation of high-speed signature schemes in the context of secure group communication.	High-speed signature schemes such as EdDSA and MQQ-SIG can build the foundation for generally applicable data origin authentication schemes, which we call unrestricted-time high-speed signing.	7, 8, 9, 10, 11, and 12.
Are subliminal channels (in high-speed signatures) a significant risk and can subliminal channels be prevented?	Theoretical analysis and practical evaluation of subliminal channels in EdDSA and MQQ-SIG signature schemes as well as theoretical evaluation of countermeasures.	Subliminal channels were found in both EdDSA and MQQ-SIG. Countermeasures are not applicable to group communication scenarios.	13
What measures are needed to secure multicast clock synchronization?	Theoretical and practical security analysis of clock synchronization protocols.	Multicast clock synchronization can be secured with unrestricted-time high-speed signing combined with sequence numbers and a novel delay measurement mitigation. We derive a formula for calculating guaranteed clock offset bounds in adversarial settings.	10

Table 1.1: Summary of research questions, methodologies, and major findings.

2
Background

This chapter has been published in part in:

Robert Annessi, Tanja Zseby, and Joachim Fabini. "A new Direction for Research on Data Origin Authentication in Group Communication". In: *International Conference on Cryptology and Network Security (CANS)*. Springer, 2017. DOI: 10.1007/978-3-030-02641-7_26. Referred to as "[I]". Adapted with permission from Springer Nature Customer Service Centre GmbH. © Springer Nature 2018.

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Robert Annessi, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252. Referred to as "[IV]". © ACM 2018.

Davor Frkat, Robert Annessi, and Tanja Zseby. "ChainChannels: Private Botnet Communication Over Public Blockchains". In: *IEEE International Conference on Blockchain (Blockchain)*. 2018. Referred to as "[V]". © IEEE 2018.

Alexander Hartl, Robert Annessi, and Tanja Zseby. "A Subliminal Channel in EdDSA: Information Leakage with High-Speed Signatures". In: *International Workshop on Managing Insider Security Threats*. MIST '17. Dallas, Texas, USA: ACM, 2017, pp. 67–78. ISBN: 978-1-4503-5177-5. DOI: 10.1145/3139923.3139925. Referred to as "[VI]". © ACM 2017.

Alexander Hartl, Robert Annessi, and Tanja Zseby. "Subliminal Channels in High-Speed Signatures". In: *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)* 9.1 (Mar. 2018), pp. 30–53. Referred to as "[VII]". © JoWUA 2018.

Robert Annessi, Joachim Fabini, Felix Iglesias, and Tanja Zseby. *Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization*. 2018. arXiv: 1811.08569 [cs.CR]. Referred to as "[VIII]".

FROM a sender-receiver multiplicity point of view transmission systems can be categorized into four classes: unicast, anycast, broadcast, and multicast. Unicast denotes communication between one sender and one receiver, anycast the communication between one sender and the closest member of a group of receivers. Broadcast and multicast both convey data from one sender to a group of receivers. Broadcast addresses all receivers (the group being potentially limited by physical constraints like network size or radio reception area) whereas multicast addresses a group of receivers that have explicitly joined a group (and therefore expressed their intent to receive data from the sender). *Group communication* comprises all one-to-many transmission systems, i.e., anycast, broadcast, and multicast.

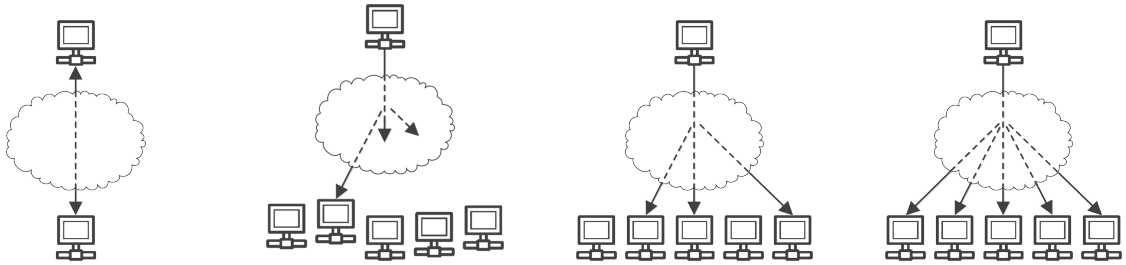

Figure 2.1: Transmission system categories.

Group communication facilitates efficient data transmission to numerous receivers and minimizes data replication. Instead of sending data to each receiver individually, data are sent just once since they are replicated by the transmission system along communications paths whenever needed. In this way, group communication is very efficient and can handle many receivers as it minimizes the number of copies that traverse the network. Group communication typically uses uni-directional data transmission over unreliable channels, because the realization of back channels and the management of replies from multiple receivers is difficult (if even feasible). Figure 2.1 summarize the basic characteristics of the four transmission system categories.

Deliberate threats to the security of information range from disgruntled employees and agents of industrial espionage to hackers, nation states and terrorists[1,2]. To mitigate those threats, goals in information security revolve around confidentiality, integrity, availability, authenticity, non-repudiation, and privacy[3]. Since not everyone is trusted (for example due to the lack of access control in many communication networks), cryptographic schemes are required to ensure that receivers can verify that messages have been, indeed, sent by the claimed sender and have not been modified along the way. This security property is called *data origin authentication*. Data origin authentication directly affects three security properties (1) integrity, (2) authenticity, and (3) non-repudiation (as depicted in Figure 2.2 on the facing page), for which we use the following definitions: Integrity (1) is a security property that enables receivers to verify whether a message has been modified during transmission in any (unknown or unauthorized) way[4,5]. Authenticity (2) is "*the property of being genuine and able to be verified and be trusted*"[6]. Two types of authentication are to be distinguished in group communication: group authentication and data origin authentication[7], which are explained in detail in Section 2.5. Non-repudiation (3) is a security property which assures that receivers can

[1] IEC/TS 62351-1:2007. *Power systems management and associated information exchange - Data and communications security – Part 1: Communication network and system security - Introduction to security issues*. IEC, Geneva, Switzerland, 2007.

[2] The Smart Grid Interoperability Panel–Smart Grid Cybersecurity Committee. *Guidelines for Smart Grid Cybersecurity*. NIST IR 7628r1. National Institute of Standards and Technology, Sept. 2014.

[3] Y. Cherdantseva and J. Hilton. "A Reference Model of Information Assurance & Security". In: *2013 Eighth International Conference on Availability, Reliability and Security (ARES)*. Sept. 2013, pp. 546–555. DOI: 10.1109/ARES.2013.72.

[4] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[5] Shirey, *Internet Security Glossary, Version 2*.

[6] Shirey, *Internet Security Glossary, Version 2*.

[7] Thomas Hardjono and Gene Tsudik. "IP multicast security: Issues and directions". In: *Annales des télécommunications*. Vol. 55. 7-8. Springer. 2000, pp. 324–340.

provide evidence (to a third-party) that a particular sender has sent a message. In this way, receivers are protected against an attempt by the sender to falsely deny sending the message[8].

[8] Shirey, *Internet Security Glossary, Version 2*.

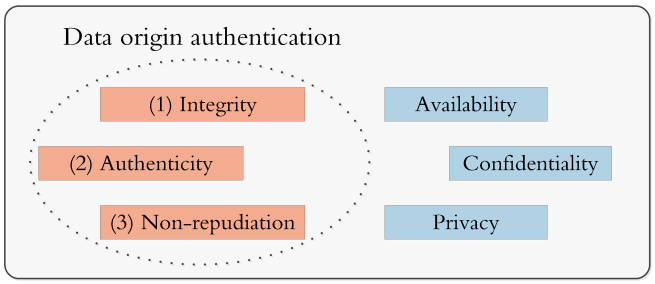

Figure 2.2: Goals in information security and their relation to data origin authentication.

For the remaining three properties we use the following definitions: (1) Confidentiality is the property that data is not disclosed to unauthorized entities.[9] (2) "*Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.*"[10] Availability (3) is the "*property of a system or a system resource being accessible, or usable or operational upon demand*".[11]

[9] Shirey, *Internet Security Glossary, Version 2*.

[10] Alan Westin. *Privacy and Freedom*. Ig Publishing, 1967. ISBN: 9781935439974.

[11] Shirey, *Internet Security Glossary, Version 2*.

2.1 Hash Functions

A hash function is a computationally efficient function that maps input of arbitrary finite length to a fixed-length output. One property of cryptographic hash functions is that their output may serve as a compact representation of the input such that it can be used as an identifier. The input of a hash function is called preimage as the output is called the image (of a particular input). Often, the output of a hash function is also referred to as message digest, hash code, hash result, hash value, or just *hash*[12].

There are three important properties of hash functions: (1) preimage resistance (or one-wayness), (2) second preimage resistance (also called universal one-way[13]), and (3) collision resistance. Preimage resistance or one-wayness (1) means that it is easy to compute the output of a hash function given the input, but it is computationally unfeasible to invert the result, i.e., find an input that produces a specific output. Second preimage resistance (2) means that given an input it is computationally unfeasible to find another input on which the hash function produces the same output. (3) Since hash functions are many-to-one relations by definition, inputs with identical output are unavoidable. Such input values with identical output are called collision. Collision resistance means that it is computationally unfeasible to find any two distinct inputs on which the hash function produces a collision, i.e., the same output[14]. Rogaway and Shrimpton discuss hash functions in greater detail[15].

[12] Menezes, Van Oorschot, and Vanstone, *Handbook of applied cryptography*.

[13] Jonathan Katz. *Digital Signatures*. Boston, MA: Springer US, 2010. ISBN: 978-0-387-27711-0.

[14] Menezes, Van Oorschot, and Vanstone, *Handbook of applied cryptography*.

[15] 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*. 3017. Springer, Feb. 5, 2004, pp. 371–388. ISBN: 978-3-540-25937-4. DOI: 10.1007/978-3-540-25937-4_24.

2.2 One-Way Chains

One-way chains (or hash chains) are an important cryptographic primitive used in many security applications such as the one-time password systems by Lamport[16] and by Haller[17,18]. The basic goal of one-way chains is to use only a single (certified) secret initially but to eventually provide a set of (certified) secrets. To this end, a (one-way) chain of secrets is generated in the following

[16] Leslie Lamport. "Password authentication with insecure communication". In: *Communications of the ACM* 24.11 (1981), pp. 770–772.

[17] Neil Haller. "The S/KEY One-Time Password System". In: *ISOC Symposium on Network and Distributed System Security*. San Diego, CA, Feb. 1994.

[18] N. Haller. *The S/KEY One-Time Password System*. RFC 1760 (Informational). RFC. Fremont, CA, USA: RFC Editor, Feb. 1995. DOI: 10.17487/RFC1760. URL: https://www.rfc-editor.org/rfc/rfc1760.txt.

way (depicted in Fig. 2.3): first, a secret s_0 is used as the first element during generation of the chain. The second element in the chain (s_1) is the hash of the first element (s_0); the third element (s_2) is the hash of the second element (s_1), and so on. The length of the chain n is fixed and represents the total number of available secrets, and the last element (s_n) is certified.

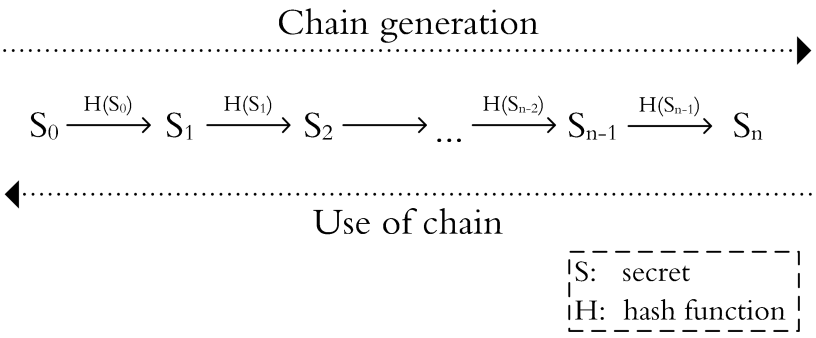

Figure 2.3: Generation and use of one-way chains.

The secrets are used in the reverse order of their generation, i.e., the last element of the chain (s_n) is used first, and the first element of the chain (s_0) is used last. The validity of a secret can be verified as soon as the subsequent secret is received, because the secret equals the hash of the subsequent secret. The first secret that is used (s_n) is certified so that the verifier can be assured that all messages are authentic and of integrity.

Both generation and verification are computationally efficient, and the preimage resistance of the hash function guarantees that the malicious generation of secrets is computationally unfeasible even when knowing previously sent secrets. At the same time, it requires computationally little effort to verify the validity of secrets.

2.3 *Message Authentication Codes*

Message Authentication Codes (MACs) are a cryptographic primitive that aims to assure authenticity and integrity of messages sent over insecure channels. MACs were first suggested by Gilbert, MacWilliams, and Sloane[19] in 1974 and further improved by Wegman and Carter[20] in 1981.

In order to assure authenticity and integrity of messages, MACs employ one-way functions and symmetric-keys. First of all, the sender generates a key *k* and shares it with all receivers in a reliable, authenticated, and confidential manner. Then, the sender constructs the authentication information of a message by passing the shared key *k* and the message *m* to a one-way function (as illustrated in step 1 of Figure 2.4 on the facing page). The authentication information is the one-way function's output h (i.e., the hash). Next (step 2), the authentication information (h) is transferred to receivers along with the message (m). The message might have been modified during transmission by unknown or unauthorized means since the communication channel is considered to be insecure. For this reason, the receiver verifies whether the message received ($m\prime$) is identical to the message (m) the sender has sent. Upon receiving a message ($m\prime$) as well as authentication information ($h\prime$), the receiver calculates the authentication information ($h\prime\prime$) using the message and

[19] Edgar N Gilbert, F Jessie MacWilliams, and Neil JA Sloane. "Codes which detect deception". In: *Bell System Technical Journal* 53.3 (1974), pp. 405–424.

[20] Mark N Wegman and J Lawrence Carter. "New hash functions and their use in authentication and set equality". In: *Journal of Computer and System Sciences* 22.3 (1981), pp. 265–279.

the shared key (step 3). In the last step (4), receivers compare the received (*h'*) to the self-calculated authentication information (*h''*). If both authentication information match, the receiver can infer that the message was received unaltered and accept the message as authentic; otherwise, the message is discarded.

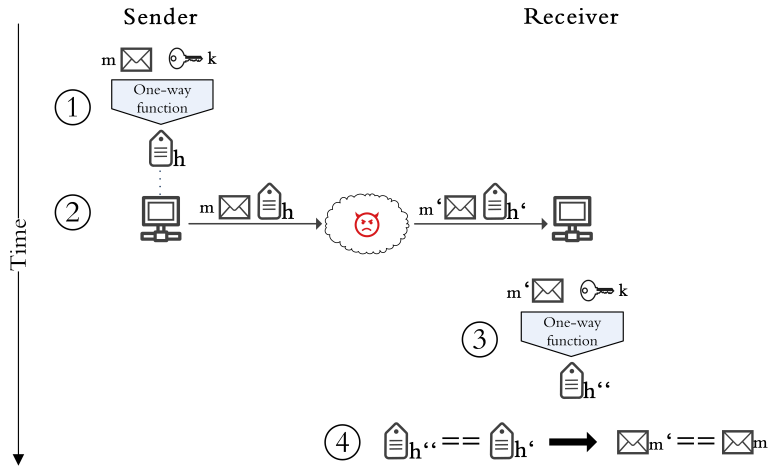

Figure 2.4: Message authentication using a shared key.

There exist well-researched, efficient, and standardized MACs based on hash functions, HMAC[21], and based on block-mode ciphers, CMAC[22]. Since efficient implementations exist for both hash functions and block-mode ciphers, MACs are widely used for unicast applications where latency and computation time are matters of concern. For group communication applications MACs have a drawback, however, since everyone who knows the shared key can generate valid MACs. This drawback will be discussed further in Section 2.5.

[21] NIST FIPS. "198: The keyed-hash message authentication code (HMAC)". in: *National Institute of Standards and Technology, Federal Information Processing Standards* (2002), p. 29.

[22] SP NIST. "800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication". In: *NIST Special Publication* (2005).

2.4 Digital Signatures

The idea of digital signatures was first suggested by Diffie and Hellman[23] in 1976. Implementations were then published by Rivest, Shamir, and Adleman[24] in 1978 and by Rabin[25] in 1979. Since then, digital signatures have become one of the most important cryptographic primitives in practice.

Like MACs, digital signatures are also concerned with the authenticity and integrity of messages. In contrast to MACs, digital signatures do not employ a single symmetric key (that is shared between all group members). Instead, the key material consists of two asymmetric parts: a *public key* that is published by the sender (i.e., shared with all receivers) and a *secret key*[26] that is kept secret by the sender. Digital signatures provide authenticity, integrity, and non-repudiation, which can be used to convince a third party that messages have not been modified or injected unnoticedly. Table 2.1 on the next page shows a brief comparison between symmetric (MACs) and asymmetric (digital) signature schemes.

A digital signature scheme consist of three functions: a key generation function, a signing function, and a verification function. The key generation function allows the sender to generate a secret signing key and a public verification key. The signing function is used by the sender to generate a signature for

[23] Whitfield Diffie and Martin E. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654.

[24] Ronald L. Rivest, Adi Shamir, and Len Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[25] Michael O. Rabin. *Digitalized signatures and public-key functions as intractable as factorization*. LCS TR-212. MIT, 1979.

[26] The secret key used for signing is often also called private key.

Scheme	Key material	Signature size	Processing speed	Security property
Symmetric	Single key	Small	Fast	Group authentication
Asymmetric	Key-pair	Large	Slow	Data origin authentication

Table 2.1: Comparison of symmetric and asymmetric signing schemes.

a particular message using the secret key. Since knowledge of the secret key is required to generate valid signatures, only the sender can compute the authentication information (assuming that the key is indeed kept secret) that can be used as a proof of origin to a third party. The receiver uses the verification function with the public key in order to validate the signature for a particular message (and therefore its authenticity). The public key allows receivers to verify the origin of the data without being able to generate valid signatures themselves.

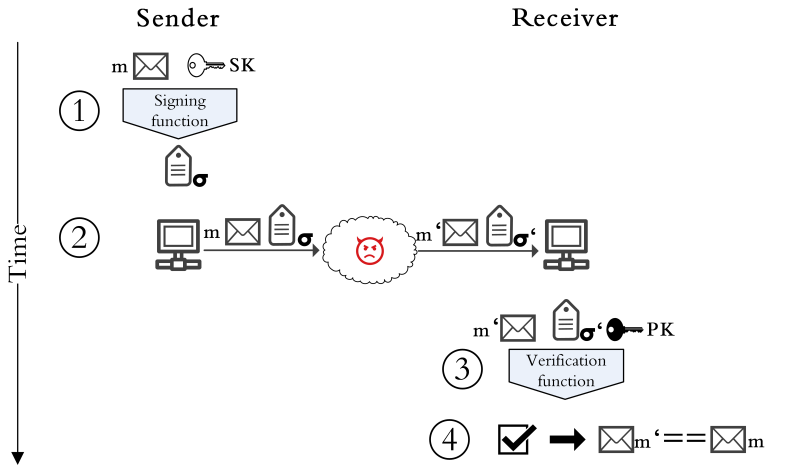

Figure 2.5: Digital signature using an asymmetric key pair.

Fig. 2.5 depicts the signing and verification of a message using an asymmetric key pair. It is assumed that the sender has already generated a key pair and shared the public key with the receiver in a reliable and authenticated (but not necessarily confidential) manner. The sender uses the secret key *SK* together with the signing function to compute the digital signature σ of message m (see step 1 in Fig. 2.5). The signature (σ) is then sent to the receivers along with the message (m) (step 2). Again, the message might have been modified by unauthorized or unknown means during transmission. Upon receiving the message ($m\prime$) and the signature ($\sigma\prime$), receivers can verify the integrity as well as the origin of the message using the verification function and the sender's public key (*PK*) (steps 3 and 4).

Usually the hash of a message is signed instead of the message itself, since the hash of a message can serve as a shorter representation of the message. By applying this hash-and-sign paradigm, the size of the signature can be reduced to a fixed length[27]. However, the collision resistance of the hash function then becomes a security requirement for the digital signature scheme[28], i.e., it must be computationally unfeasible to find two messages that produce the same hash.

[27] Oded Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. USA: Cambridge University Press, 2004. ISBN: 978-0-521-83084-3.

[28] Menezes, Van Oorschot, and Vanstone, *Handbook of applied cryptography*.

2.5 Group Authentication and Data Origin Authentication

Using the cryptographic methods described before receivers can assure that messages have been indeed sent by a legitimate sender. For group communication, two types of authentication have to be distinguished: group authentication and data origin authentication.

Group Authentication assures that data originates from a legitimate but unidentifiable group member and has not been modified by entities outside the group. MACs with a key shared by all group members are a well understood and efficient method for achieving group authentication. Nevertheless, receivers cannot distinguish between individual group members and, therefore, do not know the exact identity of the sender as all group members share the same key and can therefore generate valid MACs. This is of particular importance in group communication since there are usually many receivers involved, and a single dishonest or compromised receiver can impersonate the sender. Besides this security issue, MACs are also rather inefficient in group communication as the shared key needs to be renewed and redistributed every time a receiver leaves or joins the group. For this reason, the shared key needs to be changed frequently when groups are dynamic. In the context of group communication, the problem of group authentication is then shifted towards the problem of group key management[29].

[29] Challal, Bettahar, and Bouabdallah, "A taxonomy of multicast data origin authentication".

Many applications cannot trust all receivers and, therefore, require a level of authentication that allows receivers to identify the particular sender. *Data origin authentication* allows receivers to verify that data was indeed sent by a particular sender (non-repudiation). For data origin authentication, an asymmetric cryptographic method is required that allows receivers to verify the authenticity (of messages) without providing means to generate valid authentication information themselves on behalf of the sender. Digital signature schemes can be used to convince a third party that a message indeed originates from the claimed sender, i.e., digital signatures provide data origin authentication (in contrast to MACs). Table 2.2 summarizes the security properties provided by group authentication and by data origin authentication.

Security Property	Group Authentication	Data Origin Authentication
Integrity	✓	✓
Non-repudiation	✗	✓
Authenticity	Group	Sender

Table 2.2: Group authentication vs. data origin authentication.

The main downside of today's digital signature schemes such as RSA[30], DSA[31], and ECDSA[32] is that they come at high computational cost and therefore introduce substantial penalty in terms of delay, both in the sender and in the receiver. Consequently, it is widely believed that digital signatures are roughly 2 to 3 magnitudes slower than MACs[33] so that signing each packet is not a practical solution[34]. In this thesis, we revise the assumption that signing every packet is unfeasible as we investigate the potential of recently proposed high-performance digital signature schemes as basis for data origin authentication in group communication.

[30] Rivest, Shamir, and Adleman, "A method for obtaining digital signatures and public-key cryptosystems".

[31] Taher ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *Advances in cryptology*. Springer. 1985, pp. 10–18.

[32] Don Johnson, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)". in: *International Journal of Information Security* (2001), pp. 36–63.

[33] Katz, *Digital Signatures*.

[34] Appending a signature from a conventional signature scheme to each packet would be fine only for sporadic messages and as long as the time for signing and verification is not of concern.

In the context of signature schemes, the de facto notion of security is *existential unforgeability under adaptive chosen-message attacks*[35]. In an adaptive chosen message attack, the adversary knows the public key and can completely control what messages will be signed. Furthermore, the adversary can choose the messages to be signed depending on previous signatures that were obtained; hence the term adaptive. In our threat model (developed in Chapter 4) an adversary in control of the network can observe valid signatures from the legitimate sender. We consider this assumption to be appropriate since the adversary might even be able to influence what messages are to be signed in some applications.

Instead of requiring that it should be impossible for an attacker to forge a signature for any message at all, it should be possible only with negligible probability assuming computationally bounded resources. For this purpose, a standard security parameter $k \in N$ defines the level of security obtained by a particular instance of a scheme. The length of the public and secret keys depend on k. The larger k the larger the public and secret keys, and the less probable for an attacker to forge a valid signature. The basic security guarantee provided by a digital signature scheme (that is existentially unforgeable under adaptive chosen message attacks) is that no adversary should be able to forge a valid signature efficiently to a previously unknown message.

[35] Katz, *Digital Signatures*.

2.6 The EdDSA Signature Scheme

EdDSA[36] was introduced in 2012 by Bernstein et al. as a well performing alternative to today's signature schemes in terms of speed and security. It uses point addition on the twisted Edwards curve

$$E = \left\{(x, y) \in \boldsymbol{F}_p \times \boldsymbol{F}_p : -x^2 + y^2 = 1 + dx^2y^2\right\}, \tag{2.1}$$

where $\boldsymbol{F}_p$ denotes the Galois field of order p. The scheme has several parameters: a prime p, a parameter d defining the curve, a base point $B \in E$, the order of B denoted as L, and a cofactor 2^c with integer c such that $2^cL = |E|$ (the number of points on the curve). Furthermore, a hash function H is used that produces a $2b$ bit output, where $b \in \mathbb{N}$ determines the security level provided. These parameters are standardized for two variants: Ed25519 and Ed448[37]. Ed25519 operates on the twisted Edwards curve Curve25519[38] and yields a security level of 128 bit. Ed448 uses the Edwards Curve Curve448[39] and yields a security level of 224 bit.

The secret key k should have an entropy of at least b bit. It is mapped to a $2b$-bit string $h = H(k)$. Bits h_c to h_{n-1} with $c \leq n < b$ of h are in turn injectively mapped to a number a. Knowledge of a is enough for producing valid signatures, which justifies considering a as the signing key. The public key consists of a point on the curve $A = a \cdot B$. To generate a signature for a message M, first a nonce r is derived according to Eq. 2.2.

$$r = H(h_b, \ldots, h_{2b-1}, M) \tag{2.2}$$

A signature consists of two parts: (1) a point $R = r \cdot B$ and (2) a number $S = (r + H(R, A, M)a) \bmod L$. For verification the receiver has to check the group equation (Eq. 2.3).

$$2^cSB = 2^cR + 2^cH(R, A, M)A \tag{2.3}$$

[36] Daniel J. Bernstein et al. "High-speed high-security signatures". In: *Journal of Cryptographic Engineering* 2.2 (2012), pp. 77–89.

[37] S. Josefsson and I. Liusvaara. *Edwards-Curve Digital Signature Algorithm (EdDSA)*. RFC 8032 (Informational). RFC. Fremont, CA, USA: RFC Editor, Jan. 2017. DOI: 10.17487/RFC8032. URL: https://www.rfc-editor.org/rfc/rfc8032.txt.

[38] Daniel J. Bernstein. "Curve25519: New Diffie-Hellman Speed Records". In: *International Conference on Theory and Practice in Public-Key Cryptography (PKC)*. Springer, 2006, pp. 207–228. ISBN: 978-3-540-33852-9. DOI: 10.1007/11745853_14.

[39] Mike Hamburg. *Ed448-Goldilocks, a new elliptic curve*. Cryptology ePrint Archive, Report 2015/625. http://eprint.iacr.org/2015/625. 2015.

The EdDSA scheme is based on a digital signature scheme that was first described by Schnorr[40]. A main concern when using this kind of signatures is that the nonce r has to be unpredictable[41]. The nonce must remain secret as otherwise the signing key can be efficiently computed according to Eq. 2.4.

$$a = (S - r)/H(R, A, M) \bmod L \tag{2.4}$$

Furthermore, if identical nonces have been used for generating signatures of distinct messages M_1 and M_2, the signing key a can be found as well by Eq.2.5.

$$a = \frac{S_1 - S_2}{H(R, A, M_1) - H(R, A, M_2)} \bmod L \tag{2.5}$$

Both issues can be addressed by deriving the nonce from the message and the secret key as done for EdDSA (see Eq. 2.2). This is in contrast to ECDSA, where the issue of choosing an appropriate nonce is left to the implementation, which has to use a pseudorandom number generator or a deterministic signature generation procedure that provides cryptographically secure nonces[42].

[40] Claus P. Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *Advances in Cryptology - CRYPTO*. Springer, 1990, pp. 239–252.

[41] Bernstein et al., "High-speed high-security signatures".

[42] T. Pornin. *Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)*. RFC 6979 (Informational). RFC. Fremont, CA, USA: RFC Editor, Aug. 2013. DOI: 10.17487/RFC6979. URL: https://www.rfc-editor.org/rfc/rfc6979.txt.

2.7 MQ Signature Schemes

Digital signature schemes need to be based on problems that are mathematically hard to solve. One of these problems is solving MQ polynomials in finite fields (MQ problem). An appealing property of the MQ problem is that it allows generating signature schemes that resist quantum computer attacks.

Fig. 2.6 depicts the basic functioning of MQ signature schemes[43]. The secret key consists of two bijective affine mappings $S : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $T : \mathbb{F}_q^n \to \mathbb{F}_q^n$ and a central quadratic mapping $F : \mathbb{F}_q^m \to \mathbb{F}_q^n$. $\mathbb{F}_q$ denotes the Galois field of order q. For signing and verification a cryptographic hash function H is applied to the message, yielding the value $\boldsymbol{h} = H(M)$, with $\boldsymbol{h} \in \mathbb{F}_q^n$. In the course of signature generation the signer has to find a vector $\boldsymbol{s} \in \mathbb{F}_q^m$, such that $\boldsymbol{h} = P(\boldsymbol{s})$. In order to find such a vector, the signer first computes $\boldsymbol{y} = T^{-1}(\boldsymbol{h})$. In the next step, the signer tries to find a vector $\boldsymbol{x} \in \mathbb{F}_q^m$, for which $\boldsymbol{y} = F(\boldsymbol{x})$ holds. Since F is a quadratic function, if it consisted of polynomials with random equations, this problem would be as hard as solving $\boldsymbol{h} = P(\boldsymbol{s})$ in the first place and therefore could not be solved efficiently. F is constructed with a particular structure, however, that allows inversion straightforwardly. With knowledge of $\boldsymbol{x}$, the signer is able to find a vector $\boldsymbol{s} = S^{-1}(\boldsymbol{x})$. With knowledge of a message $\boldsymbol{m}$ and a signature $\boldsymbol{s}$, verification is straightforward. By applying $P = T \circ F \circ S$ to $\boldsymbol{s}$, the signed hash $\boldsymbol{h}$ can be computed and compared to the received hash $H(M)$.

[43] Christopher Wolf and Bart Preneel. "Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations." In: *IACR Cryptology ePrint Archive* 2005 (2005), p. 77. URL: http://eprint.iacr.org/2005/077.pdf.

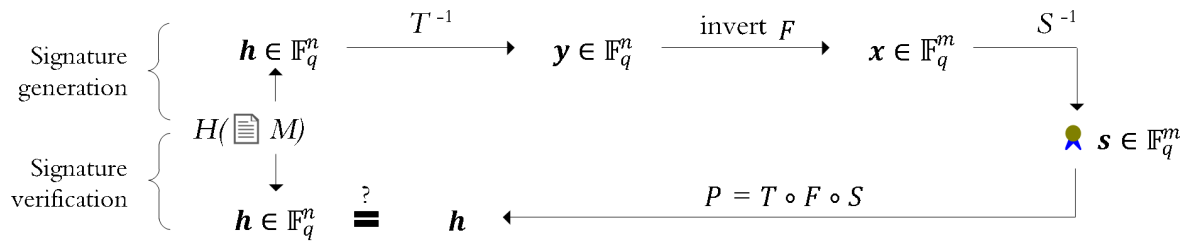

Figure 2.6: The basic operation of MQ-based signatures.

Adapted from [VII]. © JoWUA 2018.

An attacker who tries to forge a signature knows only the public key P and is confronted with the problem of finding a vector $\boldsymbol{s}$ that solves $\boldsymbol{h} = P(\boldsymbol{s})$.

If these quadratic polynomials had random coefficients, this would be hard (because of the MQ problem). The attacks developed so far for many MQ signature schemes aim to exploit the structure of the central quadratic mapping F that is meant to be hidden by the use of the two bijective affine mappings S and T. Unfortunately, all trapdoors found so far have been broken, and instead of devising completely new constructions, it has become common to modify existing (broken) schemes to yield a more secure signature scheme.

The most promising candidate for constructing a secure signature scheme is the Hidden Field Equations (HFE) trapdoor[44] in conjunction with the minus[45] and vinegar variables[46,47] modifications. HFE is a generalization of the Matsumoto-Imai scheme[48,49] that was designed after a cryptanalysis of the latter. Gligoroski, Markovski, and Knapskog proposed a trapdoor based on MQ Quasigroups (MQQ)[50,51,52]. Without using modifications all trapdoors have been broken. Wolf and Preneel give a good overview of the most important trapdoors[53].

2.8 Clock Synchronization

EACH CLOCK HAS A NATURAL DRIFT caused by the non-ideality of physical oscillators such as an oscillator's frequency affected by temperature. The aim of clock synchronization protocols is to convey time information in order to compensate for clock drift and to keep clock offsets within acceptable boundaries. Clock synchronization protocols have become an essential building block of numerous applications that rely on a precise notion of time. The deployment of clock synchronization for controlling system clocks of critical applications in telecommunication, industrial automation, financial markets, avionics, and energy distribution has increased the dependency of critical infrastructures on clocks synchronized with increasingly high precision. Examples for strict dependencies on precise time are safety-critical applications in the Smart Grid, which require a precision of 1 to 100 µs (10 µs in case of current differential line protection with high fault current sensitivity[54,55]) or MiFID II in the financial sector, requiring a precision of up to 100 µs[56,57,58]. Cellular networks also have strong requirements for synchronized clocks with $\leq 1\,\mu s$[59]. Errors in clock synchronization can lead to wrong timings and may therefore originate faulty sensor reports, endanger control decisions, and adversely affect the overall functionality of a wide range of (critical) services that depend on precise notion of time. A successful attack on clock synchronization can even undermine the security of essential cryptographic protocols[60] such as TLS.

Surprisingly, the two most widely used clock synchronization protocols, NTP[61] and PTP[62], do not provide decent security, leaving applications vulnerable to attacks. In NTP, broadcast communication is primarily used for more efficient communication because the increasing use of network-based clock synchronization results in additional load on time-servers and networks. The broadcast mode is intended for one (or a few) senders and possibly many receivers like in large corporations and institutions[63]. Many organizations operate broadcast communication infrastructure such as Ethernet or Wi-Fi, and it is convenient for them to run one (or a few) broadcast server(s) providing

[44] Jacques Patarin. "Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms". In: *Advances in Cryptology — EUROCRYPT*. Springer, 1996, pp. 33–48. ISBN: 978-3-540-68339-1.

[45] Wolf and Preneel, "Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations."

[46] Jacques Patarin. "The oil and vinegar signature scheme". In: *Dagstuhl Workshop on Cryptography*. 1997.

[47] Aviad Kipnis, Jacques Patarin, and Louis Goubin. "Unbalanced Oil and Vinegar Signature Schemes". In: *Advances in Cryptology — EUROCRYPT*. Springer, 1999, pp. 206–222. ISBN: 978-3-540-48910-8. DOI: 10.1007/3-540-48910-X_15.

[48] Hideki Imai and Tsutomu Matsumoto. "Algebraic methods for constructing asymmetric cryptosystems". In: *International Conference on Algebraic Algorithms and Error-Correcting Codes*. Springer, 1986, pp. 108–119. ISBN: 978-3-540-39855-4. DOI: 10.1007/3-540-16776-5_713.

[49] Tsutomu Matsumoto and Hideki Imai. "Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption". In: *Advances in Cryptology — EUROCRYPT: Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1988, pp. 419–453. ISBN: 978-3-540-45961-3. DOI: 10.1007/3-540-45961-8_39.

[50] Danilo Gligoroski, Smile Markovski, and Svein J. Knapskog. "A Public Key Block Cipher Based on Multivariate Quadratic Quasigroups". In: *IACR Cryptology ePrint Archive* (2008). URL: http://eprint.iacr.org/2008/320.pdf.

[51] Danilo Gligoroski, Smile Markovski, and Svein Johan Knapskog. "Multivariate Quadratic Trapdoor Functions Based on Multivariate Quadratic Quasigroups". In: *American Conference on Applied Mathematics*. MATH. Cambridge, Massachusetts: WSEAS, 2008, pp. 44–49. ISBN: 978-960-6766-47-3.

[52] Danilo Gligoroski et al. "MQQ-SIG: An Ultra-fast and Provably CMA Resistant Digital Signature Scheme". In: *International Conference on Trusted Systems*. INTRUST'11. Beijing, China: Springer, 2012, pp. 184–203. ISBN: 978-3-642-32297-6.

[53] Wolf and Preneel, "Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations."

[54] IEC/TR 61850-90-1:2010. *Communication networks and systems for power utility automation – Part 90-1: Use of IEC 61850 for the communication between substations*. IEC, Geneva, Switzerland, 2010.

[55] "IEEE Standard for Synchrophasor Measurements for Power Systems". In: *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)* (Dec. 2011), pp. 1–61. DOI: 10.1109/IEEESTD.2011.6111219.

clock synchronization to possibly many receivers[64]. In PTP, on the other hand, multicast communication is integrated into the protocol. Authenticating multicast clock synchronization, therefore, remains an interesting open problem that requires significant attention[65].

Another widely used technology for clock synchronization is the satellite-based Global Positioning System (GPS) for Pulse per Second (PPS) synchronization to global Coordinated Universal Time (UTC). The main disadvantages of GPS PPS are that it is operated by a single entity (the US Air Force)[66] and that it requires free view on at least four satellites[67], which might be difficult to get (in data centers for example). Moreover, the (public) GPS signal is not secured so that it may be spoofed with reasonable effort[68]. Due to the missing backchannel to satellites, GPS is conceptionally different to NTP and PTP. Most of the findings in this thesis related to clock synchronization, however, are applicable to other clock synchronization protocols such as GPS PPS as well.

2.8.1 The Two Phases in Clock Synchronization

Clock synchronization algorithms aim at synchronizing a slave clock to a master clock by exchanging timestamped messages over packet-switched networks. In this thesis, we assume that the master's clock is precise and reliable. Details on how the master implements and accesses such a reliable clock are considered out of this thesis' scope. Network-based clock synchronization protocols depend on two distinct phases that will be discussed below: (a) clock offset measurement and (b) delay measurement. The specific intervals of the two phases depend on the clock synchronization protocol and configuration. Delay measurements are conducted over unicast connections while clock offset measurement values are transmitted over multicast in PTP and (optionally) broadcast in NTP. Receivers correct their clock according to the transmitted timestamps and the measured delay.

Clock Offset Measurement Phase The goal of the clock offset measurement phase is to calculate the relative difference between the slave and master clocks. Clock offset can either be measured in a single message (1-step mode) supported by both NTP and PTP or in two messages (two-step mode) supported by PTP. In any case, the master sends a SYNC message to the slaves, and the slave records the transmitting timestamp of the master t_{M1}. In 1-step mode the SYNC message contains the transmitting timestamp (t_{M1}); in two-step mode the SYNC message is just used as a marker, and the FOLLOW_UP message contains the exact point in time when the SYNC message left the master (t_{M1}). This way, higher precision may be achieved because uncertainties that affect the SYNC message such as delays from the network stack can be compensated. Fig. 2.7 depicts the 2-step clock offset measurement and delay measurement.

Delay Measurement Phase SYNC and FOLLOW_UP messages are subject to various delays. Those delays are added to (and therefore negatively affect) the measured clock offset. The overall delay consists of transmission delays, queuing delays, processing delays, and propagation delays, which themselves consist of constant and stochastic parts. There is some constant delay for a given route and message size and stochastic delay that mainly depends on other traffic and

[56] Council of European Union. *Directive 2014/65/EU of the european parliament and of the council of 15 May 2014 on markets in financial instruments and amending Directive 2002/92/EC and Directive 2011/61/EU*. https://eur-lex.europa.eu/eli/dir/2014/65/oj. 2014.

[57] European Commision. *Commission delegated regulation (EU) of 7.6.2016 supplementing Directive 2014/65/EU of the European Parliament and of the Council with regard to regulatory technical standards for the level of accuracy of business clocks*. http://ec.europa.eu/finance/securities/docs/isd/mifid/rts/160607-rts-25_en.pdf. 2016.

[58] European Commision. *Annex to the Commission delegated regulation supplementing Directive 2014/65/EU of the European Parliament and of the Council with regard to regulatory technical standards for the level of accuracy of business clocks*. http://ec.europa.eu/finance/securities/docs/isd/mifid/rts/160607-rts-25-annex_en.pdf. 2016.

[59] Telecommunication standardization sector of ITU. *Time and phase synchronization aspects of telecommunication networks*. https://www.itu.int/rec/T-REC-G.8271-201708-I. 2017.

[60] David L Mills. *Computer Network Time Synchronization: the Network Time Protocol on Earth and in Space*. 2nd ed. CRC Press, 2011. ISBN: 978-1-4398-1463-5.

[61] Mills et al., *Network Time Protocol Version 4: Protocol and Algorithms Specification*.

[62] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems".

[63] Mills, *Computer Network Time Synchronization: the Network Time Protocol on Earth and in Space*.

[64] Mills, *Computer Network Time Synchronization: the Network Time Protocol on Earth and in Space*.

[65] Aanchal Malhotra and Sharon Goldberg. "Attacking NTP's Authenticated Broadcast Mode". In: *ACM SIGCOMM Computer Communication Review* 46.1 (2016), pp. 12–17.

[66] Other satellite-based systems are operated by different entities that may not necessarily be trusted either.

[67] View to four GPS satellites is the requirement for GPS-PPS signal generation in the general case. However, from a theoretical point of view, one single satellite in view is sufficient whenever the exact GPS antenna position is known

[68] Mark L. Psiaki and Todd E. Humphreys. *Protecting GPS From Spoofers Is Critical to the Future of Navigation*. https://spectrum.ieee.org/telecom/security/protecting-gps-from-spoofers-is-critical-to-the-future-of-navigation. 2016.

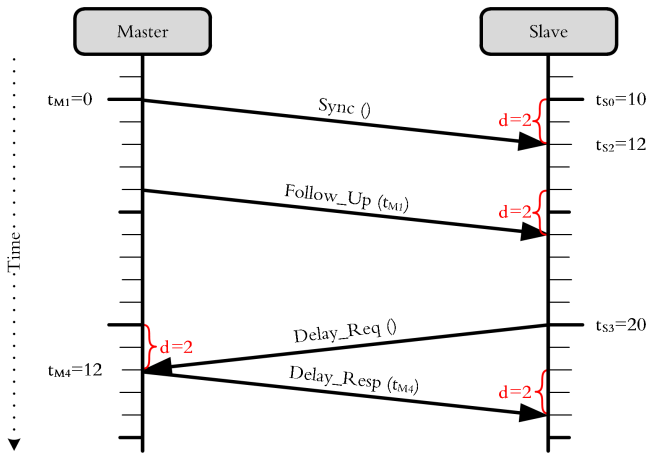

Figure 2.7: Two-step clock synchronization with deterministic and symmetric delays and initial clock offset of 10 time units.

Adapted from [VIII].

states of network devices. The goal of the delay measurement phase is to measure the overall delay and to subtract it from the measured clock offset in order to derive the actual clock offset as precisely as possible.

In PTP a delay measurement consists of two messages, one sent from the slave to the master (DELAY_REQUEST) where the slave records the transmitting time t_{S3} and a subsequent message from the master to the slave (DELAY_RESPONSE) that includes the time instant when the DELAY_REQUEST was received at the master (t_{M4}). Eventually, the slave knows four timestamps: t_{M1}, t_{S2}, t_{S3}, and t_{M4}. The slave calculates the network Round-Trip Delay (RTD) by measuring the delays in both directions (see Eq. 2.6). The RTD is calculated as the sum of the delay from master to slave ($t_{S2} - t_{M1}$) and the delay from slave to master ($t_{M4} - t_{S3}$). The One-Way Delay (OWD) from master to slave is approximated as $\frac{RTD}{2}$, assuming symmetric OWDs.

$$RTD = t_{S2} - t_{M1} + t_{M4} - t_{S3} \tag{2.6}$$

PTP allows network devices to improve the precision of clock synchronization even further by measuring and communicating the delay impaired on SYNC messages (in the corresponding FOLLOW_UP messages). Such network devices that actively support the clock synchronization protocol are called transparent clocks. Communicating the delay that transparent clocks impair on a particular message requires these devices to modify the content of the FOLLOW_UP message, which poses a challenge in the light of data origin authentication.

2.8.2 Clock Offset Calculation

In the following example we assume that the master and slave clocks have an initial clock offset of 10 time units such that the local timestamp $t_{M1} = 0$ on the master corresponds to the local timestamp $t_{S0} = 10$ on the slave (from an external observer's point of view). The OWDs are 2 time units in each direction. The slave calculates its clock offset to the master according to Eq. (2.7).

$$\mathit{offset} = t_{S2} - t_{M1} - \frac{RTD}{2} \tag{2.7}$$

Eq. (2.7) consists of the uncorrected clock offset ($t_{S2} - t_{M1}$) corrected with the OWD that is approximated by halving the RTD (Eq. 2.6). In this specific

example, the slave calculates the OWD as 2 and the clock offset as 10. Now the slave knows that its clock is 10 time units ahead of the master and can adjust accordingly. In real-world scenario, all physical clocks are subject to drift so that the process of offset correction needs to be run repeatedly in order to achieve a common notion of time.

2.9 Subliminal Communication

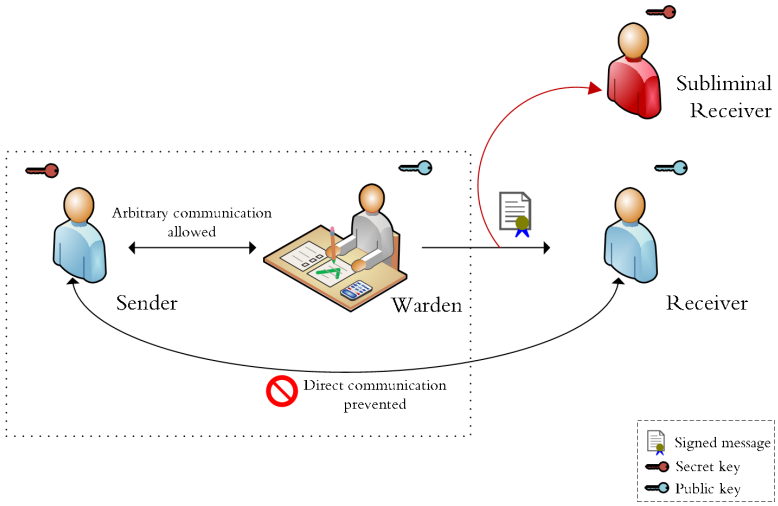

Figure 2.8: Subliminal communication in digital signatures.

Subliminal channels are hidden channels that allow unnoticeable information transmission by exploiting the mathematical structure of cryptographic schemes. In particular, digital signature schemes are an attractive candidate for being exploited as subliminal channel. In contrast to other information hiding techniques like steganography or obfuscation, the sender of the subliminal message has no need to modify the content of (overt) messages as subliminal channels are embedded in cryptographic information. Furthermore, the communication carrying subliminal information cannot be distinguished from other not carrying the channel[69].

For this reason, subliminal channels can also be used in scenarios where the signer has limited or no influence on the (overt) message to be transmitted or modifying contents would raise suspicion. A typical use of subliminal channels in digital signatures are scenarios where encryption is not permitted or is just unusual but signatures are used to ensure non-repudiation and integrity of messages.

The concept of subliminal channels was first introduced in 1993 by Simmons[70,71]. Simmons imagined two prisoners who are allowed to send each other messages[72] (see Fig. 2.8). As the warden aims to prevent the prisoners from coordinating an escape plan, only unencrypted messages are passed so that can be read. On the other hand, the prisoners' fear of the warden forging messages such that they insist on the communication being authenticated using signatures. Simmons showed that a significant part of a signature's bits can be used to leak information without giving any other but the subliminal receiver means to discover the subliminal information. In this way, a subliminal message can be embedded in a signature in such a way that the existence of the hidden message can not be detected but the signature remains verifiable.

[69] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C 20th Anniversary Edition*. John Wiley & Sons Inc, May 2015. ISBN: 978-1-119-09672-6.

[70] Gustavus J. Simmons. "Subliminal Communication Is Easy Using the DSA". in: *Advances in Cryptology — EUROCRYPT*. Lecture Notes in Computer Science. Springer, May 1993, pp. 218–232. ISBN: 978-3-540-48285-7.

[71] Gustavus J. Simmons. "The Prisoners' Problem and the Subliminal Channel". In: *Advances in Cryptology*. DOI: 10.1007/978-1-4684-4730-9_5. Springer, 1984, pp. 51–67. ISBN: 978-1-4684-4730-9.

[72] Simmons, "Subliminal Communication Is Easy Using the DSA".

2.10 Botnets

BOTNETS provide powerful infrastructures for malicious activities on the Internet. The aim of botnet operators is to produce an economically and logistically feasible, hidden, fast, and robust C&C network that is difficult to take down and to take over. In the past, the race between botnet developers and their adversaries, such as competing botnet operators or authorities, led to highly innovative and sophisticated C&C infrastructures[73]. Nevertheless, the main weak point and leverage against botnets often turned out to be a vulnerability in the C&C concept, which could be used for detection and take-downs of botnets[74,75]. The infrastructure is the key to the robustness of a botnet against takedown or takeover.

The main terms used in the context of botnets are:

- The *botmaster* controls the botnet and aims to stay stealthy.
- The *bots* reside on infected hosts and use their resources to perform tasks as commanded by the botmaster such as stealing private information or performing Distributed Denial of Service (DoS) (DDoS) attacks.
- The *C&C infrastructure* is used to send commands to the bots. It can be centralized, a P2P, or a hybrid structure and may use simple Internet relay chat (IRC) channels, HTTP(S), or more sophisticated neoteric protocols[76].

[73] G. Vormayr, T. Zseby, and J. Fabini. "Botnet Communication Patterns". In: *IEEE Communications Surveys Tutorials* 19.4 (2017), pp. 2768–2796.

[74] C. Czosseck, G. Klein, and F. Leder. "On the Arms Race around Botnets - Setting up and Taking down Botnets". In: *International Conference on Cyber Conflict*. June 2011, pp. 1–14.

[75] Y. Nadji, R. Perdisci, and M. Antonakakis. "Still Beheading Hydras: Botnet Takedowns Then and Now". In: *IEEE Transactions on Dependable and Secure Computing* 14.5 (Sept. 2017), pp. 535–549.

[76] Vormayr, Zseby, and Fabini, "Botnet Communication Patterns".

3
Related Work

This chapter has been published in part in:

Robert Annessi, Tanja Zseby, and Joachim Fabini. "A new Direction for Research on Data Origin Authentication in Group Communication". In: *International Conference on Cryptology and Network Security (CANS)*. Springer, 2017. DOI: 10.1007/978-3-030-02641-7_26. Referred to as "[I]". Adapted with permission from Springer Nature Customer Service Centre GmbH. © Springer Nature 2018.

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Davor Frkat, Robert Annessi, and Tanja Zseby. "ChainChannels: Private Botnet Communication Over Public Blockchains". In: *IEEE International Conference on Blockchain (Blockchain)*. 2018. Referred to as "[V]". © IEEE 2018.

Alexander Hartl, Robert Annessi, and Tanja Zseby. "A Subliminal Channel in EdDSA: Information Leakage with High-Speed Signatures". In: *International Workshop on Managing Insider Security Threats*. MIST '17. Dallas, Texas, USA: ACM, 2017, pp. 67–78. ISBN: 978-1-4503-5177-5. DOI: 10.1145/3139923.3139925. Referred to as "[VI]". © ACM 2017.

Alexander Hartl, Robert Annessi, and Tanja Zseby. "Subliminal Channels in High-Speed Signatures". In: *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)* 9.1 (Mar. 2018), pp. 30–53. Referred to as "[VII]". © JoWUA 2018.

Robert Annessi, Joachim Fabini, Felix Iglesias, and Tanja Zseby. *Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization*. 2018. arXiv: 1811.08569 [cs.CR]. Referred to as "[VIII]".

3.1 Data Origin Authentication for Group Communication

UNTIL 2003, surveys[1,2,3,4] did not focus on data origin authentication specifically but addressed security issues in group communication generally. This changed, however, as Cucinotta, Cecchetti, and Ferraro compared four different authentication schemes[5]. Challal, Bettahar, and Bouabdallah provided a comprehensive overview and taxonomy of data origin authentication schemes[6]. Their publication is from 2004, however, so that it does not cover any later developments that were published since then. Seys and Preneel compared ECDSA, Lamport-Diffie, and Hash to Abtain Random Subsets (HORS) regarding energy efficiency[7] in 2005. In 2008, He et al. published a notably short survey on data origin authentication[8]. Kaur, Sangal, and Kumar provided a brief survey on MAC-based authentication schemes[9] in 2012. In 2013, Law et al. compared four authentication schemes (BiBa, TV-HORS, SCU+, and TSV+) with application to Smart Grids[10]. In 2015, Grover and Lim evaluated authentication schemes regarding their applicability to wireless sensor networks[11].

There is a clear need for a comprehensive survey on data origin authentication schemes for group communication that evaluates their general applicability. As described, however, related work is either dated, focuses on specific applications or aspects only, or compares just few data origin authentication schemes. In contrast to related work, this thesis classifies and evaluates the last twenty-five years of research on data origin authentication for general group communication. We suggest a new classification of data origin authentication schemes that also covers research developments in recent years (Chapter 7) and conduct a comprehensive evaluation (Chapter 8) based on criteria that result from our threat analysis (Chapter 4) and requirements (Chapter 6) for the particular use cases under test (Chapter 5).

3.2 Secure Clock Synchronization

SECURITY MEASURES for clock synchronization can either be implemented as an integrated part of clock synchronization protocols or through external security mechanisms, such as (D)TLS[12,13] or IPsec[14] that are (somewhat) independent. There are three main drawbacks associated with external security mechanisms, in particular with (D)TLS and IPsec. First, they lower the precision of clock synchronization[15] because of the delays and the delay variation in the security stack. Second, (D)TLS and IPsec can provide only group authentication for group communication but not data origin authentication because both employ shared symmetric keys. Third, they prevent intermediate devices in PTP (transparent clocks) from adding information, which would improve the precision of clock synchronization by communicating the delay introduced by that devices to clock synchronization messages. Another external security measure is MACsec, which is lim-

[1] Judge and Ammar, "Security issues and solutions in multicast content distribution".

[2] M.J. Moyer, J.R. Rao, and P. Rohatgi. "A survey of security issues in multicast communications". In: *IEEE Network* 13.6 (Nov. 1999), pp. 12–23. ISSN: 0890-8044. DOI: 10.1109/65.806981.

[3] Hardjono and Tsudik, "IP multicast security: Issues and directions".

[4] Thomas Hardjono, Lakshminath R. Dondeti, and Radia Perlman. *Multicast and Group Security*. USA: Artech House, Inc., 2003. ISBN: 1580533426.

[5] Tommaso Cucinotta, Gabriele Cecchetti, and Gianluca Ferraro. "Adopting redundancy techniques for multicast stream authentication". In: *IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS)*. 2003.

[6] Challal, Bettahar, and Bouabdallah, "A taxonomy of multicast data origin authentication".

[7] Stefaan Seys and Bart Preneel. "Power consumption evaluation of efficient digital signature schemes for low power devices". In: *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob)*. Vol. 1. IEEE, 2005, pp. 79–86.

[8] Jinxin He et al. "Survey on multicast data origin authentication". In: *IEEE International Conference on Communication Technology (ICCT)*. Nov. 2008, pp. 749–752. DOI: 10.1109/ICCT.2008.4716234.

[9] Ramanpreet Kaur, Amrit Lal Sangal, and Krishan Kumar. "Mac based multicast source authentication: A survey". In: *International Journal of Computer Application* 37.2 (2012).

[10] Yee Wei Law et al. "Comparative Study of Multicast Authentication Schemes with Application to Wide-area Measurement System". In: *ACM SIGSAC Symposium on Information, Computer and Communications Security*. ASIACCS. 2013, pp. 287–298. ISBN: 978-1-4503-1767-2. DOI: 10.1145/2484313.2484349.

[11] Kanika Grover and Alvin Lim. "A survey of broadcast authentication schemes for wireless networks". In: *Ad Hoc Networks* 24 (Jan. 2015), pp. 288–316. ISSN: 15708705. DOI: 10.1016/j.adhoc.2014.06.008.

[12] E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Aug. 2018. DOI: 10.17487/RFC8446. URL: https://www.rfc-editor.org/rfc/rfc8446.txt.

[13] E. Rescorla and N. Modadugu. *Datagram Transport Layer Security Version 1.2*. RFC 6347 (Proposed Standard). RFC. Updated by RFCs 7507, 7905. Fremont, CA, USA: RFC Editor, Jan. 2012. DOI: 10.17487/RFC6347. URL: https://www.rfc-editor.org/rfc/rfc6347.txt.

[14] S. Frankel and S. Krishnan. *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*. RFC 6071 (Informational). RFC. Fremont, CA, USA: RFC Editor, Feb. 2011. DOI: 10.17487/RFC6071. URL: https://www.rfc-editor.org/rfc/rfc6071.txt.

[15] Albert Treytl, Bernd Hirschler, and Thilo Sauter. "Secure tunneling of high-precision clock synchronization protocols and other time-stamped data". In: *IEEE International Workshop on Factory Communication Systems (WFCS)*. IEEE, 2010, pp. 303–312.

ited to OSI layer 2 and can therefore only provide security to Local Area Networks (LANs). Furthermore, each involved node needs to be trusted as MACsec follows a hop-by-hop encryption approach.

PTP includes an experimental security extension, Annex K[16], that provides message integrity and replay protection. Annex K is based on symmetric-key cryptography, however, and therefore cannot provide data origin authentication. Furthermore, several flaws were discovered and Annex K was never properly formalized[17,18]. NTP, on the other hand, incorporates two integrated security mechanisms to provide authenticity and integrity: symmetric[19] and Autokey[20]. Both mechanisms can only achieve group authentication for broadcast communication but not data origin authentication either. Furthermore, the use of Autokey is strongly discouraged as severe security weaknesses of the algorithm have been discovered[21,22]. As potential successors of Autokey two distinct proposals have been suggested: ANTP[23] and Network Time Security (NTS)[24,25,26]. ANTP is intended to provide security for unicast NTP only, and cannot be extended to provide efficient data origin authentication for broadcast clock synchronization as it is based on symmetric-key cryptography. NTS consist of a set of Internet Engineering Task Force (IETF) drafts that aim to provide authenticity and integrity for unicast and multicast clock synchronization protocols. During this thesis, the IETF NTP working group removed securing multicast clock synchronization as a goal for NTS, since they want to focus on securing unicast clock synchronization first (before securing multicast clock synchronization). Until now, NTS is only specified for NTP. A formal analysis was conducted for the unicast mode[27] but no evaluation is provided to motivate the use of Timed Efficient Stream Loss-Tolerant Authentication (TESLA)[28] as favorite candidate for securing multicast clock synchronization. TESLA is highly rated for securing also the next version of PTP[29]. Besides other existing data origin authentication schemes, TESLA will be evaluated in Chapter 8 for its suitability to secure multicast clock synchronization. We are the first to conduct a comprehensive evaluation of the suitability of data origin authentication schemes to secure clock synchronization protocols.

A recently suggested security extension to PTP[30,31] aims to secure the 2-step mode in PTP. It provides data origin authentication using Ed25519 to prevent substitution attacks, and provides replay protection with sequence numbers. However, since the authors use multiple threat models and did not conduct a comprehensive security analysis, their sequence number window approach facilitates DoS attacks were an adversary just needs to inject SYNC messages with the highest acceptable sequence number. In this way, no other valid messages will be accepted by the receiver anymore. Furthermore, their proposal is susceptible to pre-play attacks because sequence numbers are predictable and there is no link between SYNC and FOLLOW_UP messages other than those predictable sequence numbers. In this way, an adversary can make receivers adhere to a false time by pre-playing SYNC messages. Delay attacks, on which we focus on specifically, are only men-

[16] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems".

[17] A. Treytl and B. Hirschler. "Security flaws and workarounds for IEEE 1588 (transparent) clocks". In: International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS). Oct. 2009, pp. 1–6. DOI: 10.1109/ISPCS.2009.5340204.

[18] N. Moreira et al. "Security mechanisms to protect IEEE 1588 synchronization: State of the art and trends". In: IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS). Oct. 2015, pp. 115–120. DOI: 10.1109/ISPCS.2015.7324694.

[19] Mills et al., *Network Time Protocol Version 4: Protocol and Algorithms Specification*.

[20] B. Haberman (Ed.) and D. Mills. *Network Time Protocol Version 4: Autokey Specification*. RFC 5906 (Informational). RFC. Fremont, CA, USA: RFC Editor, June 2010. DOI: 10.17487/RFC5906. URL: https://www.rfc-editor.org/rfc/rfc5906.txt.

[21] Stephen Röttger. "Analysis of the NTP Autokey Procedures". Master's thesis, Technische Universitt Braunschweig, 2012.

[22] Harlan Stenn. *[ntpwg] Antw: Re: Proposed REFID changes*. http://lists.ntp.org/pipermail/ntpwg/2015-July/002291.html. [Online; accessed 12-September-2018]. 2015.

[23] Benjamin Dowling, Douglas Stebila, and Greg Zaverucha. "Authenticated Network Time Synchronization". In: *USENIX Security Symposium*. Austin, TX, Aug. 2016, pp. 823–840. ISBN: 978-1-931971-32-4.

[24] Dieter Sibold, Stephen Roettger, and Kristof Teichel. *Network Time Security*. Internet-Draft draft-ietf-ntp-network-time-security-11. Oct. 2015. URL: http://www.ietf.org/internet-drafts/draft-ietf-ntp-network-time-security-11.txt.

[25] Dieter Sibold, Stephen Roettger, and Kristof Teichel. *Using the Network Time Security Specification to Secure the Network Time Protocol*. Internet-Draft draft-ietf-ntp-using-nts-for-ntp-02. Oct. 2015. URL: http://www.ietf.org/internet-drafts/draft-ietf-ntp-using-nts-for-ntp-02.txt.

[26] Dieter Sibold et al. *Protecting Network Time Security Messages with the Cryptographic Message Syntax (CMS)*. Internet-Draft draft-ietf-ntp-cms-for-nts-message-06. Feb. 2016. URL: https://tools.ietf.org/html/draft-ietf-ntp-cms-for-nts-message-06.

[27] Kristof Teichel, Dieter Sibold, and Stefan Milius. "First Results of a Formal Analysis of the Network Time Security Specification". In: *Security Standardisation Research*. Lecture Notes in Computer Science 9497. Springer, 2015, pp. 218–245. ISBN: 978-3-319-27152-1.

[28] A. Perrig et al. *Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction*. RFC 4082 (Informational). RFC. Fremont, CA, USA: RFC Editor, June 2005. DOI: 10.17487/RFC4082. URL: https://www.rfc-editor.org/rfc/rfc4082.txt.

[29] Moreira et al., "Security mechanisms to protect IEEE 1588 synchronization".

tioned in the extended version of their paper but Itkin and Wool do not propose a solution; instead they refer to related work. Moreover, the authors falsely assume that signing breaks hardware timestamping, which is only true for the one-step mode but not for the two-step mode (as to be shown in Chapter 10).

Delay Attacks

Tsang and Beznosov were the first to analyze the security of PTP in 2005[32,33]. They describe delay attacks briefly and propose to average delay measurements as a countermeasure. We argue that averaging delay measurements also averages the malicious delay such that the attack is only barely mitigated. The attack has even full effect as soon as the duration of the attack is longer than the averaging interval[34]. The authors furthermore propose to check for abnormal clock offset values. But abnormal values (i.e., spikes in the measured clock offset) only occur for simple delay attacks, not for incremental delay attacks (as to be shown in Section 10.1.1).

While Annex K does not cover attacks in the time domain at all, the PTP standard[35] mentions asymmetric one-way delays (although in a single sentence only) but does not cover delay attacks any further. NTS covers delay attacks briefly. The authors of the security extension to PTP[36,37] mention delay attacks in the extended version of their paper but refer to related work when it comes to countermeasures. We analyze asymmetric OWD attacks in Section 10.1.3 and countermeasures against such attacks in Section 10.1.4.

Ellegaard as well as Koskiahde, Kujala, and Norolampi recommend MACsec to secure clock synchronization[38,39]. Mizrahi analyzes IPsec and MACsec as means to secure clock synchronization[40]. Delay attacks are very briefly mentioned but there are neither details on how delay attacks can be conducted or mitigated nor how IPsec and MACsec relate to delay attacks. Treytl and Hirschler discuss the usage of IPsec to protect PTP[41] but do not mention delay attacks at all. Mizrahi states that encrypting clock synchronization traffic makes it more difficult to conduct delay attacks[42]. We show that this holds true to some extent for selective messages attacks only (Section 10.1.1), but asymmetric OWD attacks (Section 10.1.3) are not obstructed by encryption at all

Yang, An, and Yu simulated delay attacks and propose a countermeasure based on hypothesis testing[43]. Moreira et al. discuss delay attacks briefly[44]. Ullmann and Vögeler examined delay attacks on PTP and NTP[45] and propose what to limit the maximum RTD as mitigation. Lisova et al. provide a good analysis of the consequences of delay attacks and discusses limiting the maximal RTD as countermeasure[46]. Other countermeasures discussed, such as monitoring interarrival times, are unreasonable in a general setting because those times will be equally affected by a delay attack and cannot be used therefore as a countermeasure against delay attacks. Mizrahi also briefly discusses limiting the maximal RTD as a potential countermeasure against delay attack[47].

[30] E. Itkin and A. Wool. "A security analysis and revised security extension for the precision time protocol". In: *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. Sept. 2016, pp. 1–6. DOI: 10.1109/ISPCS.2016.7579501.

[31] Eyal Itkin and Avishai Wool. "A Security Analysis and Revised Security Extension for the Precision Time Protocol". In: *CoRR* abs/1603.00707 (May 28, 2016). URL: http://arxiv.org/abs/1603.00707.

[32] Jeanette Tsang and Konstantin Beznosov. *A Security Analysis of the Precise Time Protocol*. LERSSE technical report, Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada, LERSSE-TR-2006-02, 2005.

[33] Jeanette Tsang and Konstantin Beznosov. "A Security Analysis of the Precise Time Protocol (Short Paper)". In: Springer, 2006, pp. 50–59.

[34] And if the averaging interval is too long, the clock synchronization precision may be negatively affected (since reacting on changes in the RTD would take long as well).

[35] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems".

[36] Itkin and Wool, "A security analysis and revised security extension for the precision time protocol".

[37] Itkin and Wool, "A Security Analysis and Revised Security Extension for the Precision Time Protocol".

[38] L. Ellegaard. *PTP Security using MACsec*. Tech. rep. https://ieee-sa.centraldesktop.com/1588/file/33390811/. P1588 Working Group, Aug. 2014.

[39] T. Koskiahde, J. Kujala, and T. Norolampi. "A sensor network architecture for military and crisis management". In: *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. Sept. 2008, pp. 110–114. DOI: 10.1109/ISPCS.2008.4659223.

[40] T. Mizrahi. "Time synchronization security using IPsec and MACsec". In: *2011 International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*. Sept. 2011, pp. 38–43. DOI: 10.1109/ISPCS.2011.6070153.

[41] A. Treytl and B. Hirschler. "Securing IEEE 1588 by IPsec tunnels - An analysis". In: *International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*. Sept. 2010, pp. 83–90. DOI: 10.1109/ISPCS.2010.5609765.

[42] Tal Mizrahi. "A game theoretic analysis of delay attacks against time synchronization protocols". In: *International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*. IEEE, 2012, pp. 1–6.

[43] Qingyu Yang, Dou An, and Wei Yu. "On time desynchronization attack against IEEE 1588 protocol in power grid systems". In: *IEEE Energytech*. May 2013, pp. 1–5. DOI: 10.1109/EnergyTech.2013.6645332.

[44] Moreira et al., "Security mechanisms to protect IEEE 1588 synchronization".

[45] M. Ullmann and M. Vögeler. "Delay attacks - Implication on NTP and PTP time synchronization". In: *International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. Oct. 2009, pp. 1–6. DOI: 10.1109/ISPCS.2009.5340224.

Mizrahi proposes to use multiple paths between master and slave to mitigate delay attacks[48]. The assumptions are quite strong, however, as all paths need to be entirely (i.e., also physically) independent and the adversary may only attack a minority of paths successfully. From a practical perspective it seems unrealistic to provide multiple low-latency paths with low delay variation that are entirely independent. It is implicitly assumed that the various networking components such as Network Interface Cards (NICs), routers, and switches are from distinct vendors and that paths are symmetric. Furthermore, the countermeasure does not scale well as it is more costly to establish new independent paths than to compromise any majority of paths.

Research on secure clock synchronization often either excludes delay attacks or refers to related work for discussion on this issue. None of the related work discusses delay attacks on encrypted traffic. Furthermore, related work does not identify the fundamental limitation inherent to network-based clock synchronization: clock synchronization can either be high-precision or secure against delay attacks.

3.3 Subliminal Communication

SIMMONS showed how to construct narrowband channels that allow transmitting a few bits as well as broadband channels that allow a significant amount of subliminal information to be added to a signature[49]. Such broadband channels require the receiver of the hidden information to know (parts of) the signer's secret key in order to recover the subliminal information. Noteworthy in this context is the Newton channel[50] that was found by Anderson et al. for the ElGamal signature scheme[51] specifically. When using the Newton channel, the signer unveils as many bits of information about the secret key to the subliminal receiver as should be usable subliminal bandwidth. So far, subliminal channels have been shown to exist in many traditional signature schemes such as DSA[52,53], ECDSA[54,55,56] or RSA[57,58,59], and finding a mode of operation that is provably subliminal-free often turns out to be difficult.

The concept of subliminal channels is related to Secretly Embedded Trapdoor with Universal Protection (SETUP) attacks that were introduced by Young and Yung[60]. When performing a SETUP attack, an adversary replaces a cryptographic algorithm on a victim's device by an altered algorithm, aiming to break its security. In the context of digital signatures this means that the modified signing algorithm leaks the secret key to the adversary. Young and Yung also introduced the term "Kleptography", which is defined as the *"study of stealing information securely and subliminally"*[61]. Recently, attacks based on modifying cryptographic algorithms attract a new research interest and are now called Algorithm-Substitution Attacks (ASAs)[62,63] and Subversion Attacks (SAs)[64].

[46] Elena Lisova et al. "Protecting Clock Synchronization: Adversary Detection through Network Monitoring". In: *Journal of Electrical and Computer Engineering* 2016 (2016), pp. 1–13. ISSN: 2090-0147, 2090-0155. DOI: 10.1155/2016/6297476. URL: http://www.hindawi.com/journals/jece/2016/6297476/ (visited on 02/08/2017).

[47] Mizrahi, "A game theoretic analysis of delay attacks against time synchronization protocols".

[48] Mizrahi, "A game theoretic analysis of delay attacks against time synchronization protocols".

[49] Simmons, "Subliminal Communication Is Easy Using the DSA".

[50] Ross Anderson et al. "The Newton channel". In: *Information Hiding: First International Workshop*. Springer, 1996, pp. 151–156. ISBN: 978-3-540-49589-5. DOI: 10.1007/3-540-61996-8_38.

[51] ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms".

[52] Simmons, "Subliminal Communication Is Easy Using the DSA".

[53] ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms".

[54] Jens-Matthias Bohli, Maria Isabel Gonzalez Vasco, and Rainer Steinwandt. "A subliminal-free variant of ECDSA". in: *International Workshop on Information Hiding*. Springer, 2006, pp. 375–387.

[55] Q. Dong and G. Xiao. "A Subliminal-Free Variant of ECDSA Using Interactive Protocol". In: *International Conference on E-Product E-Service and E-Entertainment*. Nov. 2010, pp. 1–3. DOI: 10.1109/ICEEE.2010.5660874.

[56] Johnson, Menezes, and Vanstone, "The elliptic curve digital signature algorithm (ECDSA)".

[57] Xianfeng Zhao and Ning Li. "Reversible Watermarking with Subliminal Channel". In: *International Workshop on Information Hiding*. Springer, 2008, pp. 118–131. ISBN: 978-3-540-88961-8. DOI: 10.1007/978-3-540-88961-8_9.

[58] Jens-Matthias Bohli and Rainer Steinwandt. "On Subliminal Channels in Deterministic Signature Schemes". In: *International Conference on Information Security and Cryptology (ICISC)*. Springer, 2005, pp. 182–194. ISBN: 978-3-540-32083-8. DOI: 10.1007/11496618_14.

[59] Rivest, Shamir, and Adleman, "A method for obtaining digital signatures and public-key cryptosystems".

[60] Adam Young and Moti Yung. "The Dark Side of "Black-Box" Cryptography or: Should We Trust Capstone?" In: *Advances in Cryptology*. Springer, 1996, pp. 89–103. ISBN: 978-3-540-68697-2. DOI: 10.1007/3-540-68697-5_8.

[61] Young and Yung, "The Dark Side of "Black-Box" Cryptography or: Should We Trust Capstone?"

[62] Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. "Security of Symmetric Encryption against Mass Surveillance". In: *Advances in Cryptology*. Springer, 2014, pp. 1–19. ISBN: 978-3-662-44371-2. DOI: 10.1007/978-3-662-44371-2_1.

[63] Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. "Subversion-Resilient Signature Schemes". In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Denver, Colorado, USA, 2015, pp. 364–375. ISBN: 978-1-4503-3832-5.

Data Insertion in Blockchains

BLOCKCHAINS were initially used for cryptocurrencies. More applications developed on top of them such as time stamping and notary services, but also malicious and illicit content emerged[65]. The Bitcoin blockchain was extensively analyzed for the insertion of arbitrary data[66,67]. It was shown that the OP_RETURN field and payments to fake public key, key hashes, or script hashes can be used to inject data. The authors show that as long as the transactions are valid, arbitrary content can be injected into blockchains. Nevertheless, the content is transmitted overtly and can be observed by anyone.

Ali et al. showed that the Bitcoin blockchain is viable as C&C infrastructure, using key leakage, the OP_RETURN field, and a narrowband subliminal channel based on brute-forcing during signature generation[68,69] . The authors point out that C&C communication over the Bitcoin network inherits its key strengths: the blockchain provides low latency, is decentralized by design, has a consistent network state, and that it is hard to censor C&C instructions without significantly impacting the overall function of the blockchain. Also, the proposed communication channel scales well with the number of bots and offers the botmaster an infrastructure that is very unlikely to be taken down and therefore less risky and less costly to operate.

Zombiecoin[70] is tied to the Bitcoin blockchain. Zombiecoin uses the OP_RETURN field and a narrowband subliminal channel that is computationally expensive (and not practical to use). In contrast, we suggest *ChainChannels* (Chapter 13) that is based on a key leakage method and a broadband subliminal channel. Computational requirements in ChainChannels are low and the entire nonce can be used to transmit subliminal information. ChainChannels does not require any specific blockchain field because it relies solely on one thing that all blockchains have in common: digital signatures.

[64] Ateniese, Magri, and Venturi, "Subversion-Resilient Signature Schemes".

[65] Arvind Narayanan et al. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, July 2016. ISBN: 978-0-691-17169-2.

[66] Andrew Sward, Ivy Vecna, and Forrest Stonedahl. "Data Insertion in Bitcoin's Blockchain". In: *Ledger* 3.0 (Apr. 2018).

[67] Roman Matzutt et al. "A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin". In: *International Conference on Financial Cryptography and Data Security*. Springer, 2018, p. 18.

[68] Syed Taha Ali et al. "ZombieCoin 2.0: Managing next-Generation Botnets Using Bitcoin". In: *International Journal of Information Security* (June 2017), pp. 1–12.

[69] Syed Taha Ali et al. "ZombieCoin: Powering Next-Generation Botnets with Bitcoin". In: *Financial Cryptography and Data Security*. Lecture Notes in Computer Science. Springer, Jan. 2015, pp. 34–48. ISBN: 978-3-662-48051-9.

[70] Ali et al., "ZombieCoin".

4
Threat Model

This chapter has been published in part in:

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Robert Annessi, Joachim Fabini, Felix Iglesias, and Tanja Zseby. *Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization*. 2018. arXiv: 1811.08569 [cs.CR]. Referred to as "[VIII]".

In this chapter, we present the network model, the adversary model, as well as the application model that we consider most appropriate in the context of group communication. We are confident that the models are valid and do not reduce the generality of the results. The models are used throughout the thesis (and we note any deviation). Table 4.1 shows a brief summary.

Model	Properties
Adversary	full control of network; can compromise receivers; computational power limited but not bounded to that of the sender or receivers.
Network	untrusted; unreliable
Application	messages not known in advance and sent at different, constant or variable rates.

Table 4.1: Summary of the threat model.

4.1 Network Model

We consider group communication involving one sender and a potentially large number of receivers. Messages are delivered from the sender to each receiver through an unreliable, potentially lossy communication network. The network only forwards packets and does not provide any security guarantee such as integrity, authenticity, or availability. For this reason, the communication channel is considered insecure[1], which means that packets may be dropped, read, modified, or injected by parties other than those for which the

[1] Menezes, Van Oorschot, and Vanstone, *Handbook of applied cryptography*.

information is intended. This corresponds to the Dolev-Yao threat model[2]. Furthermore, we assume that neither network devices nor receivers are trusted since the larger the number of receivers, the higher the probability that at least one is dishonest or compromised. Besides, receivers may not be controlled by the same entity that controls the sender.

[2] Danny Dolev and Andrew C Yao. "On the security of public key protocols". In: *IEEE Transactions on Information Theory* 29.2 (1983), pp. 198–208.

4.2 Adversary Model

The goal of the adversary is to conduct one or more of the attacks described in Section 4.4. We assume that the adversary has full control over the network and can selectively read, drop, resend, reorder, inject, delay, and alter packets arbitrarily with negligible delay[3]. Although the adversary could potentially drop all packets, we assume that the adversary does not prevent the communication between sender and receiver permanently. I.e., the receiver will receive at least some packets, but we do not make any assumptions about the actual ratio of received to sent messages. The computational power of the adversary is limited but not necessarily bounded to that of the sender nor that of the receivers, i.e., the adversary can use computationally more powerful devices and larger storage. The adversary can compromise an arbitrary number of receivers, can learn all their secrets and may even impersonate them. An important aspect of assuming a powerful adversary is that if security properties hold against such powerful adversary, they also hold against less capable adversaries. We are confident that this approach provides better long-term results than suggesting solutions against less capable adversaries and then adapting those solutions towards stronger adversaries.

[3] Such powerful adversary could emerge from a privileged Man in the Middle (MITM) position in the network, by having gained access to a network node or a link of the communication path or by conducting an Address Resolution Protocol (ARP) poisoning attack for example.

4.3 Application Model

We consider communications involving one sender and a potentially large number of receivers, and applications that generate messages not known to the sender in advance, i.e., before they are to be sent. Additionally, applications may send messages at various, constant or variable rates. We assume that receivers already know the (long-term) public key of the particular sender and can trust that this public key is both valid and correctly bound to the sender's identity (we assume that the public key is certified and has been transmitted initially in a reliable and authenticated manner), i.e., the initial key exchange and the certification problems are solved. Key distribution can be conducted securely in the receiver's initial setup phase or provided during runtime by a certificate authority, for example.

4.4 Attacks

The goal of the adversary is to attack the communication between sender and receivers. To this end, the adversary may conduct the following attacks.

Message Removal Attack A message removal attack results when a message is intentionally dropped by the adversary. As previously stated, we assume that the adversary does not drop all messages because this would then be an issue of availability rather than authenticity and integrity, and data origin authentication schemes cannot prevent such message removal attack.

Message Injection Attack In a message injection attack, the adversary injects a forged message in order to deceive receivers such that the receivers accept the message as if it had been sent by the claimed sender.

Message Substitution Attack In a substitution attack[4], the adversary intercepts a valid message during transmission and aims to modify it, such that receivers accept the forged message as if it had been sent by the original sender. Message substitution attacks can be seen as a combination of a message removal attack and a message injection attack, in which the message injection attack uses information from the captured message.

[4] Josef Pieprzyk, Jennifer Seberry, and Thomas Hardjono. *Fundamentals of Computer Security*. USA: Springer, 2002. ISBN: 3540431012.

Message Replay Attack An adversary can record messages and replay them without modification at a later time, since successful verification of a message does not certify the correctness of the message's send time[5]. In this way, inaccurate information can be intentionally provided to receivers. It is, therefore, important for authentication schemes to protect against replay attacks by ensuring that every message is distinct and can be tied to a specific session or time.

[5] Katz, *Digital Signatures*.

Message Pre-play Attack In contrast to replay attacks, the adversary injects messages at an earlier point in time in pre-play attacks, before the sender has even sent the message. For this purpose, the adversary needs to be able to predict future messages.

Sender Impersonation Attack For conducting an impersonation attack[6], the adversary utilizes its knowledge of the authentication scheme used by the sender. By constructing authentication information along with forged messages, the adversary attempts to fool receivers into thinking that those messages were sent by the claimed sender. In this way, the adversary impersonates the sender in order to distribute false information to receivers.

[6] Pieprzyk, Seberry, and Hardjono, *Fundamentals of Computer Security*.

Message Delay Attack To conduct a message delay attack, an adversary intercepts a message and delays that message artificially for some time before forwarding it. Some applications can be manipulated by choosing the effective delay maliciously. Unfortunately, data origin authentication schemes cannot prevent adversaries from conducting message delay attacks, as successful verification of messages does not imply that send and propagation times along the network path are correct[7].

[7] Katz, *Digital Signatures*.

Message Flooding In DoS attacks[8], the computational or storage capacities of receivers are to be exhausted in order to prevent or delay the reception of valid messages. Such DoS attacks can be conducted, for example, by an adversary sending an excessive number of messages to a receiver. Data origin authentication schemes cannot prevent such message flooding attacks, but they can reduce their impact as they can provide means for receivers to distinguish valid from invalid messages (up to a certain number of packets per time interval).

[8] Shirey, *Internet Security Glossary, Version 2*.

5
Use Cases

This chapter has been published in part in:

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Robert Annessi, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252. Referred to as "[IV]". © ACM 2018.

Robert Annessi, Joachim Fabini, Felix Iglesias, and Tanja Zseby. *Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization*. 2018. arXiv: 1811.08569 [cs.CR]. Referred to as "[VIII]".

5.1 Clock Synchronization

For clock synchronization, we abstract the multicast clock synchronization scenario from any specific implementation such as NTP or PTP whenever possible[1]. In multicast communication a sender periodically sends clock synchronization messages that can be received by possibly many receivers. The goal of the adversary is to make receivers adhere to a false time, degrade the precision of clock synchronization, or deny access to the clock synchronization service. Our aim is to maintain security as well as the lowest possible degradation of clock synchronization's precision. In particular, the time that is needed for signing packets and verifying signatures can be detrimental to the precision of clock synchronization protocols.

Data origin authentication can not or only partially prevent message flooding and message removal attacks. Those attacks degrade precision only, however, and we argue that it is the task of an Intrusion Detection System (IDS) or other dedicated components to defend against such attacks. An attacker who has full control of the network (MITM) or can perform sufficiently powerful DoS attacks can always prevent clock synchronization to happen in the

[1] Details on delay attacks in Section 10.1 are described in terms of PTP but the main results apply to other clock synchronization protocols as well.

first place. Henceforth, we will focus on preventing attacks that could make receivers synchronize to a false time and therefore could cause serious disturbance of applications that rely on a precise notion of time: substitution attacks, replay attacks, pre-play attacks, and delay attacks.

Table 5.1 summarizes the threat analysis for clock synchronization. The impact column provides a measure of the attack's severity, which may have two values: (1) false time and (2) degraded precision. False time (1) means that receivers may synchronize to a false time due to the attack, which can cause a disturbance of applications that rely on a precise notation of time. Degraded precision (2) means that the attack causes clock synchronization of receivers to be degraded. As pointed out, attacks that just degrade precision, message removal and message flooding, can not or only partially be prevented by data origin authentication. Also, additional measures are required in order to mitigate delay attacks. Nevertheless, two severe attacks (message substitution and sender impersonation) can be prevented entirely by data origin authentication, which is also a prerequisite to prevent replay attacks. In the following, we list the properties that authentication schemes need to satisfy in order to be suitable to secure multicast clock synchronization.

Attack	Impact	Prevented by or with data origin authentication
Message substitution	False time	✓ Yes
Message replay	False time	✓ Yes
Sender impersonation	False time	✓ Yes
Message removal	Degraded precision	✗ No
Message delay	False time	✗ No
Message flooding	Degraded precision	✗ Partially

Table 5.1: Clock synchronization threat analysis.

Adapted from [II]. © IEEE 2017.

The following properties are important for data origin authentication schemes in order to secure multicast clock synchronization:

Computational Efficiency The generation as well as verification of authentication information should require as low computational effort as possible to minimize the impairment onto the precision of clock synchronization[2,3,4].

Immediate Signing and Immediate Verification Some data origin authentication schemes require the sender or the receiver to buffer packets before the authentication information can be generated or verified, respectively. In this way, schemes can reduce the overall computational resources needed for signing and verification. For sender-side buffering, additional delay is introduced, which may lower the precision of clock synchronization. When receivers need to buffer messages, an adversary can flood the receivers' buffers with bogus messages. In this way, a DoS attack can be conducted, because receivers are unable to store and authenticate incoming messages when their buffers are

[2] Treytl, Hirschler, and Sauter, "Secure tunneling of high-precision clock synchronization protocols and other time-stamped data".

[3] Malhotra and Goldberg, "Attacking NTP's Authenticated Broadcast Mode".

[4] Dowling, Stebila, and Zaverucha, "Authenticated Network Time Synchronization".

full. Depending on the data origin authentication scheme, sender-side and receiver-side buffering may apply to the same packet.

Message Replay Protection There are various ways to perform replay attacks, and no single measure can prevent them generally. In multicast clock synchronization, however, replay protection can be ensured by making every message distinct. As an example, one way could be adding a cryptographic nonce such as a monotonically increasing number[5] and receivers (as well as the sender) keeping track of the nonce last used.

[5] Katz, *Digital Signatures*.

Robustness to Packet Loss Clock synchronization protocols are often built upon communication networks that provide best-effort delivery of messages and cannot guarantee reliability. Many data origin authentication schemes, however, do not take the lossy nature of these networks into account as they require the underlying communication networks to provide reliability.

Independence of Clock Synchronization Some data origin authentication schemes require receivers' clocks to be synchronized with the sender's. This is a critical requirement, since an authentication scheme depending on clock synchronization leads to a circular dependency in the context of clock synchronization protocols. Such circular dependency introduces an additional attack vector because the security of the authentication scheme breaks if the underlying assumption of synchronized sender and receiver clocks is violated. However, the dependency between authentication scheme and time cannot be avoided entirely since the lifetime of cryptographic keys must be enforced, for example. Nevertheless, this dependency introduces a security risk, and therefore should be as loose as possible (as in the case of NTP 34 or 68 years[6]).

[6] Mills et al., *Network Time Protocol Version 4: Protocol and Algorithms Specification*.

5.2 5G Networks

5G NETWORKS offer new communication possibilities for critical infrastructures. In 5G networks, it is paramount to use the available resources efficiently considering the increasing use of high-bandwidth applications over cellular networks. Therefore, group communication becomes essential in order to distribute data to groups of receivers efficiently without unnecessary data duplication and transmission as the demand for group communication services over cellular networks is rapidly increasing. And this trend is expected to continue for 5G networks.

Group communication services in 5G networks can be classified either as human-oriented (1) or as machine-oriented (2). Human-oriented services (1) are, for example, enhanced TV-services such as (high-definition) audio and video downloading, streaming, or distribution, news or advertising services that can be enhanced by grouping users based on their interest, preferences, or other characteristics. Other human-oriented services include location-based services ranging from augmented reality services that allow users to receive additional information from the surrounding environment such as for visitors in a city, or public safety services such as Mission Critical Push to Talk (MCPTT) used by police officers, fire fighters, or train operation personnel[7], or disas-

[7] J. Kim et al. "Group communication over LTE: a radio access perspective". In: *IEEE Communications Magazine* 54.4 (Apr. 2016), pp. 16–23. ISSN: 0163-6804. DOI: 10.1109/MCOM.2016.7452261.

ter recovery where users receive emergency information. Considering the huge number of connected sensors and machines expected over 5G networks, machine-oriented services (2) such as smart homes, smart industrial plants, intelligent transportation systems, or software updates also become apparent[8].

In this thesis, we address the security of group communication services in 5G networks in terms of authenticity and integrity. Security is crucial to the successful deployment and future use of 5G networks—not only because network operators are concerned about attacks harming their reputation and revenue, but also because impersonation of group communication services may lead anywhere from inconveniences to catastrophes (depending on the particular service). Security measures for group communication services are essential as industry and critical infrastructures increasingly depend on cellular networks and future 5G infrastructures. In this thesis, we analyze whether group communication services in 5G networks are appropriately secured in terms of authenticity and integrity, and we evaluate data origin authentication schemes to improve security for group communication in 5G networks.

[8] Giuseppe Araniti et al. "Multicasting over Emerging 5G Networks: Challenges and Perspectives". In: *IEEE Network* 31.2 (Mar. 2017), pp. 80–89. ISSN: 0890-8044. DOI: 10.1109/MNET.2017.1600067NM.

Security Measures Related to Group Communication in 5G

In cellular networks, User Equipment (UE) such as mobile phones or mobile network devices that perform their action without human assistance (i.e., machines) are connected to Evolved Node Bs (eNBs) over radio. eNBs are enhanced base stations that incorporate all radio interface related functions. For group communication, eNBs receive data over the core network from the Broadcast Multicast - Service Centre (BM-SC). The BM-SC receives the content from the actual content provider over a unicast connection, but practically acts as source of the group communication service. The communications between the BM-SC and the eNBs as well as the communication between the eNB and the UEs are conducted over multicast and broadcast interfaces, respectively. Fig. 5.1 on the following page depicts group communication services over 5G and the related security measures. In the following, we will briefly describe these security measures.

Content Provider to BM-SC (1) The connection between content provider and BM-SC usually is unicast and therefore out of this thesis' scope. Conventional security measures such as IPsec or TLS can provide integrity, authenticity, and confidentiality and therefore appropriate security for this connection.

BM-SC to eNB (2) The communication between BM-SC and eNB can be secured with IPsec or TLS as long as it is conducted over a unicast connection. IPsec security in terms of authenticity, integrity, and (optionally) confidentiality is mandatory between the edges of networks operated by distinct administrative authorities, which basically provides a hop-by-hop security approach in which securing the communication is optional within a network (according to the standard)[9]. It needs to be stressed that IPsec can only secure unicast connections - unless Group Domain of Interpretation (GDOI)[10] is used, which itself can provide only group authentication but not data origin authentication. For group communication, the security entirely depends on the group communication specific security measures (described later in this subsection).

[9] 3GPP. *TS 33.210 Network Domain Security (NDS); IP network layer security, Rel. 14*. Dec. 2016.

[10] B. Weis, S. Rowles, and T. Hardjono. *The Group Domain of Interpretation*. RFC 6407 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Oct. 2011. DOI: 10.17487/RFC6407. URL: https://www.rfc-editor.org/rfc/rfc6407.txt.

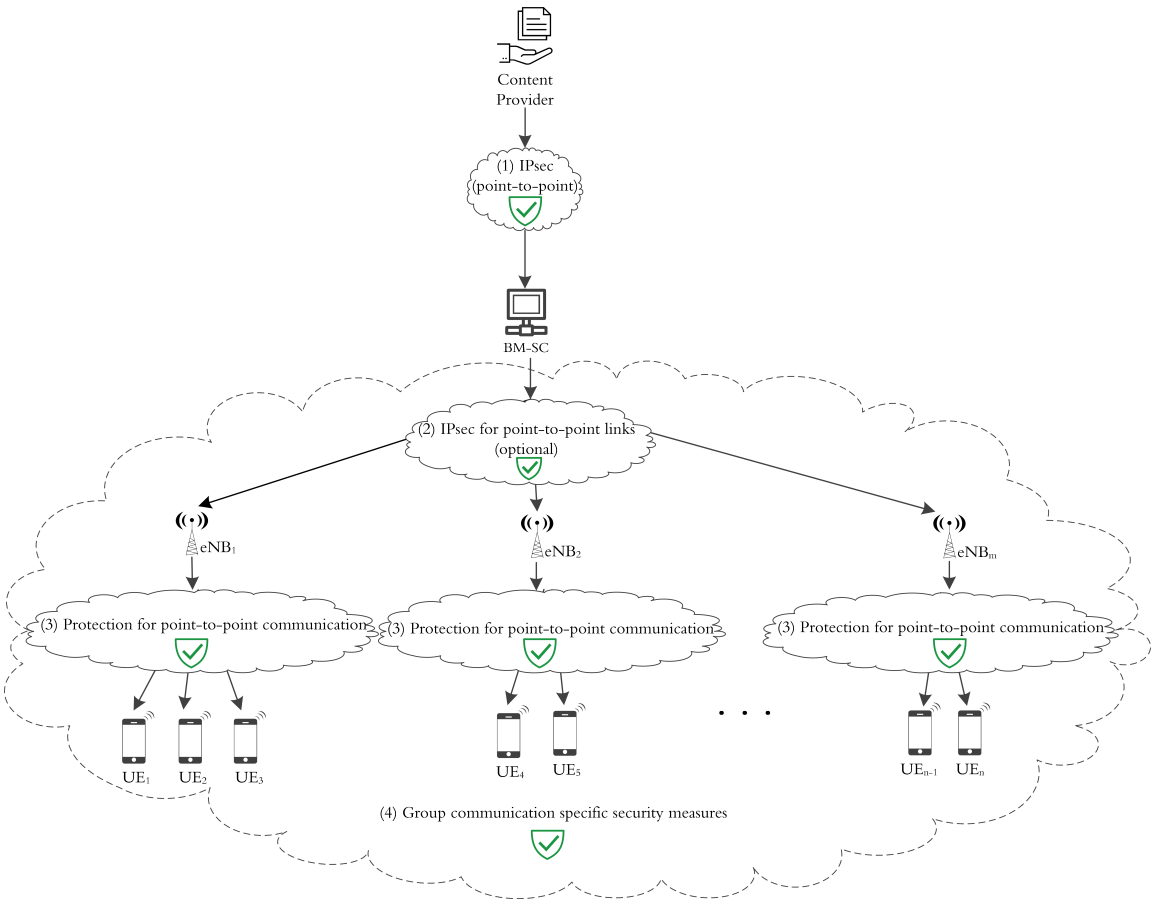

Figure 5.1: Security measures related to group communication in 5G.

Adapted from [IV]. © ACM 2018.

eNB to UE (3) The integrity of the communication from eNBs to UEs is commonly protected in terms of integrity, confidentiality, and authenticity. This protection between eNB and UE is achieved as data are scrambled with UE-specific keys such that only the UE can decode the data[11]. However, this protection is only provided for unicast but not for group communication[12]. For this reason, the security of group communication entirely depends on the group communication specific security measures (described next).

Group Communication Specific Security Measures (4) Datagrams sent from the BM-SC (over eNBs) to UEs should be protected from eavesdroppers that are not allowed to receive the data as well as from adversaries that aim to modify or inject datagrams maliciously. Since the conventional security measures outlined before in this section cannot provide end-to-end security for group communication from the BM-SC to the UEs, additional end-to-end protection methods were introduced both for the Multimedia Broadcast/Multicast Service (MBMS)[13] and for Single Cell-Point To Multipoint (SC-PTM)[14]. MBMS and SC-PTM are radio access methods that provide group communication functionality. SC-PTM is optimized for mid-size groups while MBMS is optimized for large groups. In this way, both methods are complementary. These protection methods were designed to provide end-to-end integrity, authenticity, and confidentiality for group communication. The group communication specific security measures included in MBMS and SC-PTM differ but do have in common that they employ a symmetric group key to secure

[11] 3GPP. *TS 33.401 3GPP System Architecture Evolution (SAE), Rel. 15*. June 2017.

[12] 3GPP. *TS 33.246 Security of Multimedia Broadcast/Multicast Service (MBMS), Rel. 14*. Dec. 2016.

[13] 3GPP. *TS 33.246 Security of Multimedia Broadcast/Multicast Service (MBMS), Rel. 14*.

[14] Kim et al., "Group communication over LTE".

communications[15]. The group keys are shared between the BM-SC and every UE that has access to the particular group communication service.

As highlighted in Chapter 2, security measures based on symmetric keys can only provide group authentication, and therefore neither MBMS nor SC-PTM provide data origin authentication. This shortcoming of the group communication specific security measures in MBMS and SC-PTM together with the conventional security measures that are not applicable to group communication become important in the attack scenarios presented in Chapter 11 (when the benignity or operational security of the network operator or UEs cannot be trusted).

[15] In case of MBMS the symmetric group key is called MBMS Traffic Key (MTK), and for SC-PTM it is called Radio Network Temporary Identifier (RNTI).

5.3 Sensor Data Collection in Smart Grids

Smart Grids enhance electrical grids with information technology to optimize grid operation. Power grids have to meet highest requirements in availability and quality of supply. As a consequence high requirements for availability, integrity, authenticity and possibly also confidentiality, have to be considered for Smart Grids. Digital signatures are therefore of fundamental importance for providing authentication, and the use of high-speed signatures is favorable in many situations when signatures should be processed on low-power hardware like sensor devices, if a large amount of data has to be signed, or if low latency is important.

Phasor Measurement Units (PMUs) measure the phasor of electrical current and voltage and transmit the measurements to Phasor Data Concentrators (PDCs), supporting control decisions for grid operation (see Fig. 13.2). The measurements have to be transmitted in real-time, where up to 60 to 120 measurements are performed each second. Authenticity and integrity is crucial as otherwise wrong control decisions might result.

6
Requirements

This chapter has been published in part in:

Robert Annessi, Tanja Zseby, and Joachim Fabini. "A new Direction for Research on Data Origin Authentication in Group Communication". In: *International Conference on Cryptology and Network Security (CANS)*. Springer, 2017. DOI: 10.1007/978-3-030-02641-7_26. Referred to as "[I]". Adapted with permission from Springer Nature Customer Service Centre GmbH. © Springer Nature 2018.

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Robert Annessi, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252. Referred to as "[IV]". © ACM 2018.

In this chapter, we discuss how the threats identified in Chapter 4. Complementing the main security goals (i.e., authenticity, integrity, and non-repudiation) a set of requirements for data origin authentication schemes is defined in this chapter that enable receivers to detect injected or manipulated messages. The variety of contexts in which group communication applications operate leads to various additional constraints. Based on our threat model (Chapter 4) and use cases (Chapter 5), we derive requirements for data origin authentication schemes in order to evaluate their suitability to secure group communication. Overall, we distinguish three categories: performance, security, and robustness. Table 6.1 on the next page summarizes the requirements.

6.1 Performance

Computational Efficiency The creation as well as the verification of authentication information should need only minor computational effort, so that authentication schemes can be used for real-time applications and can also be implemented within resource constrained devices. Furthermore, if a group

Performance	Security	Robustness
Computational efficiency	Collusion resistance	Robustness to packet loss
Low communication overhead	Quantum computer resistance	Resilience against DoS attacks
Immediate signing	Information-theoretical security	Independence of clock synchronization
Immediate verification		Support for a wide range of applications
Scalability		Secure channel only initially required
Dynamic membership handling		Robustness to subliminal channels

Table 6.1: Evaluation criteria for data origin authentication schemes in group communication.

consists of many members, differences in computational resources and storage capacities among group members are more likely. Since overall performance in group communication is limited by the weakest member, it is important that the authentication schemes need little computational resources; otherwise, the performance may need to be decreased for the whole group.

Low Communication Overhead As bandwidth is a worthy resource, communication overhead should be low. Furthermore, since bandwidth resources may not be evenly distributed among group members as some group members may have high-bandwidth while others have low-bandwidth connections, the authentication information should be as small as possible. Also, the bandwidth required to distribute the key material to receivers should be small.

Immediate Signing Some schemes require the sender to buffer messages before the authentication information can be created. In this way, schemes reduce the computational resources needed for signing and verification. Such buffering introduces additional delay, however, that might be intolerable to certain types of applications, such as real-time applications. For this reason, data origin authentication schemes should provide immediate signing.

Immediate Verification Other schemes require receivers to buffer messages before they can be authenticated. Such buffering introduces additional delay and also increase the risk of DoS attacks as buffers may be filled with bogus messages by an attacker. For this reason, authentication schemes should provide immediate verification.

Scalability Since group communication, in general, is a model of one-to-many communication, data origin authentication schemes must scale well with the group size. Otherwise, the benefits of group communication are effectively negated.

Dynamic Membership Handling Dynamic groups, in which members freely join and leave the group, can have a significant impact on the efficiency of the authentication scheme. In some applications, like cellular networks for example, membership may depend on the location of the receivers which is subject to constant change.

6.2 Security

Collusion Resistance Since receivers are not (necessarily) trusted, data origin authentication schemes must provide protection against receivers that collude in order to impersonate the sender. However, some authentication schemes are parameterized on the number of colluders they can resist.

Quantum Computer Resistance In 1994, Shor showed that quantum computers would be eventually able to factor big integers efficiently[1]. For this reason, quantum computers will break conventional public-key cryptography as soon as quantum computers become sufficiently large and stable. Since lots of research is conducted regarding universal quantum computers that can run Shor's algorithm[2] efficiently, "*it now appears that quantum computers are feasible, and will be implemented at some point in the future.*"[3]. Although sufficiently large and stable quantum computers do not exist as yet, we consider them as potential future threat since systems are often designed to operate over time periods of 10, 20, or even 30 years[4,5]. Data origin authentication schemes should therefore not only be secure today but also remain secure long-term against future attacks that employ quantum computers. However, schemes that provide quantum computer resistance today comprise significant drawbacks so that they are only employed when necessary. In the context of secure group communication, there is more time with authenticity than with confidentiality, which needs to hold for many years or even decades. Authenticity in group communication needs to hold for roughly the duration of the communication only. For this reason, we consider quantum computer resistance as appealing property of data origin authentication schemes but not as an absolute requirement (as long as post-quantum secure schemes can be used as a drop-in replacement without requiring significant effort).

Information-Theoretical Security In conditional security, the computational and time resources of an adversary are assumed to be bounded. A data origin authentication schemes is considered information-theoretical secure[6], if the computational and time resources of the adversary are not assumed to be bounded.

6.3 Robustness

Robustness to Packet Loss In general, reliable transmission requires a back-channel from the receiver to the sender. Such back-channels cannot be assumed to be generally available in group communication because the underlying communication network (such as satellite communication) may not provide such back-channel. Furthermore, most group communication applications use communication networks that provide best-effort delivery of data only and cannot guarantee reliability (and reliable group communication is a distinct research area[7]). If a packet gets lost during transmission, each receiver will ask the sender for retransmission and the sender may be overwhelmed by the number of requests. Many data origin authentication schemes, however, do not take the lossy nature of those networks into account. Often, data origin authentication schemes require some form of reliability from the underlying

[1] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Symposium on Foundations of Computer Science*. Nov. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[2] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM J. Comput.* 26.5 (Oct. 1997), pp. 1484–1509. ISSN: 0097-5397. DOI: 10.1137/S0097539795293172. URL: http://dx.doi.org/10.1137/S0097539795293172.

[3] Ralph Merkle. *Comments in 2012 about the 1979 paper: A Certified Digital Signature*. http://www.merkle.com/papers/Certified1979.pdf. [Online; accessed 08-August-2018]. 2012.

[4] Dong Wei et al. "Protecting Smart Grid Automation Systems Against Cyberattacks". In: *IEEE Transactions on Smart Grid* 2.4 (Dec. 2011), pp. 782–795. ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2159999.

[5] T. Zseby. "Is IPv6 Ready for the Smart Grid?" In: *International Conference on Cyber Security*. Dec. 2012, pp. 157–164. DOI: 10.1109/CyberSecurity.2012.27.

[6] Information-theoretical security is often also referred to as unconditional security.

[7] A. Popescu et al. "A Survey of Reliable Multicast Communication". In: *Next Generation Internet Networks*. May 2007, pp. 111–118. DOI: 10.1109/NGI.2007.371205.

transmission system. The extent of reliability that is required can vary from receiving the majority of packets to in-sequence ordering of packets without any packet loss. It is important that data origin authentication schemes provide authenticity of received packets in case other packets were lost. If packets were lost and the authentication scheme fails to verify the authenticity of received packets, it is said to be degrading[8].

[8] Christophe Maurice Andre Tartary. *Authentication for Multicast Communication*. Macquarie University, 2007.

Resilience Against DoS Attacks Data origin authentication schemes must not introduce additional attack vectors themselves that facilitate DoS attacks.

Independence of Clock Synchronization Some schemes require sender and receiver clocks to be synchronized. This is a drawback to particular applications such as clock synchronization, since depending on clock synchronization introduces an additional prerequisite and may also add a potential attack vector.

Support for a Wide Range of Applications Some schemes are tied to particular application scenarios and make strong assumptions such as the number of messages in a specific time interval or that the sender must know the messages to be sent in advance; other schemes require a particular network topology for example. Data origin authentication schemes should be applicable to a wide range of potential applications, however.

Secure Channel Only Initially Required As stated in Chapter 4, we assume that the initial key distribution problem is solved so that receivers know the certified public key of the sender. We do not assume, however, a secure out-of-band channel to be available for key distribution permanently during the communication.

Robustness to Subliminal Channels While subliminal channels are researched for digital signature schemes, they have not been researched particularly in the context of data origin authentication for group communication. However, when employing data origin authentication on a large scale (or in high-speed environments), the problem of subliminal channels becomes prevalent. For this reason, we address subliminal channels in high-speed signature schemes in Chapter 13.

7

Classification of Data Origin Authentication Schemes

This chapter has been published in part in:

Robert Annessi, Tanja Zseby, and Joachim Fabini. "A new Direction for Research on Data Origin Authentication in Group Communication". In: *International Conference on Cryptology and Network Security (CANS)*. Springer, 2017. DOI: 10.1007/978-3-030-02641-7_26. Referred to as "[I]". Adapted with permission from Springer Nature Customer Service Centre GmbH. © Springer Nature 2018.

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Robert Annessi, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252. Referred to as "[IV]". © ACM 2018.

Data origin authentication schemes for group communication have matured for more than twenty-five years, and many ideas were proposed to solve this challenging problem. Due to the sheer number of schemes that have been proposed, an appropriate classification is essential to provide an overview. In this chapter, we introduce a new classification of data origin authentication schemes for group communication (see Figure 7.1 on the facing page). We identify three conceptional distinct approaches. The first approach aims to extend symmetric schemes to data origin authentication. The other two approaches aim to overcome the computational intensive nature of public-key based authentication schemes: reducing the cost of conventional signatures schemes and designing fast authentication schemes.

Building upon the classification[1] by Challal, Bettahar, and Bouabdallah, we suggest seven distinct classes of data origin authentication schemes: secret-information asymmetry, deferred signing[2], signature propagation, signature dispersal, One-Time Signature (OTS), Multiple-Time Signature (MTS), and our new unrestricted-time high-speed signing class. Schemes from the secret-

[1] Challal, Bettahar, and Bouabdallah, "A taxonomy of multicast data origin authentication".

[2] Challal, Bettahar, and Bouabdallah originally used the term "*differed signing*", but we think that they actually meant "*deferred signing*" as it makes more sense in this context.

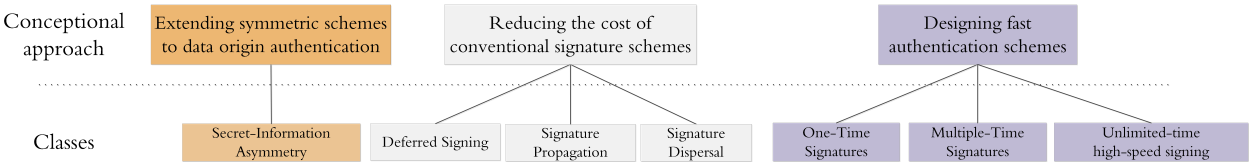

Figure 7.1: Classification of data origin authentication schemes for group communication.

information asymmetry class follow the approach of extending symmetric schemes to data origin authentication. Schemes from the deferred signing, signature propagation, and signature dispersal classes follow the approach of reducing the computational cost of conventional signature schemes. Schemes from the OTS, MTS, and unrestricted-time high-speed signing classes follow the approach of designing fast authentication schemes. In this chapter, we introduce all seven classes of data origin authentication schemes as well as their corresponding three approaches. We are confident that this classification is more generally suitable, especially considering developments in recent years.

7.1 Extending Symmetric Schemes to Data Origin Authentication

The first approach (extending symmetric schemes to data origin authentication) is followed by one class of schemes: secret-information asymmetry.

7.1.1 Secret-Information Asymmetry

Secret-information asymmetry was first suggested by Simmons[3] in 1988 and later proposed for group communication by Desmedt, Frankel, and Yung[4] in 1992. With secret-information asymmetry schemes, such as k-MAC[5], the sender shares a set of keys with receivers instead of just a single key. The sender knows the entire set of keys and therefore can generate valid authentication information but each receiver's partial view (of the keys) allows just to verify but not to generate authentication information. The k-MAC scheme uses distinct keys to calculate receiver-specific MACs. Then, all MACs are appended to a packet. Upon reception of the packet, each receiver can verify the MACs it has the keys for but cannot create valid authentication information on behalf of the sender as the other keys are unknown. In this way, symmetric keys are extended to data origin authentication.

[3] G. J. Simmons. "A survey of information authentication". In: *Proceedings of the IEEE* 76.5 (May 1988), pp. 603–620. ISSN: 0018-9219. DOI: 10.1109/5.4445.

[4] Y. Desmedt, Y. Frankel, and M. Yung. "Multi-receiver/multi-sender network security: efficient authenticated multicast/feedback". In: *Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM)*. May 1992, 2045–2054 vol.3. DOI: 10.1109/INFCOM.1992.263476.

[5] R. Canetti et al. "Multicast Security: A Taxonomy and Some Efficient Constructions". In: Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). vol. 2. Mar. 1999, pp. 708–716. DOI: 10.1109/INFCOM.1999.751457.

7.2 Reducing the Cost of Conventional Signature Schemes

The next approach is to reduce the computational cost of conventional signature schemes. Three different classes of data origin authentication schemes can be distinguished that aim to reduce the computational cost of conventional signature schemes: (1) deferred signing, (2) signature propagation, and (3) signature dispersal. Each class will be described briefly in the following.

7.2.1 Deferred Signing

With schemes from the deferred signing class, such as offline/online signing[6], the signing process is split into two steps: a slow offline and a fast online step. In the online step, each packet is signed using a OTS scheme[7], which is com-

[6] Shimon Even, Oded Goldreich, and Silvio Micali. "On-line/off-line digital signatures". In: *Journal of Cryptology* 9.1 (1996), pp. 35–67.

[7] One-time signature schemes will be explained in detail in Sections 7.3.1 and 8.3.1.

putationally very efficient. To ensure that the one-time keys originate from the claimed sender, a (conventional) digital signature scheme with a certified public key is used in the offline step to sign every one-time key. The slow part, the generation and signing of the one-time keys, is independent of the actual packet to be signed and, therefore, can be conducted offline in advance. High performance is achieved because the online step consists of signing messages with computationally very efficient OTSs. The deferred signing class is useful in settings, in which online response time is more important than offline processing time.

7.2.2 *Signature Propagation*

Another method to reduce the computational cost of conventional signatures is followed by signature propagation schemes such as Receiver driven Layer Hash-chaining (RLH)[8], where the computational cost of signing and verifying is amortized over multiple packets. In schemes from the signature propagation class, a signature from a conventional signature scheme is appended to just one packet, the *signature packet*. The hashes of non-signature packets are included in preceding packets so that they can be verified through later packets. In this way, a chain of packets is built in which each packet carries the hash of the subsequent packet. Only the first packet in the chain is signed and contains the hash of the second packet. In this way, the signature propagates through all packets so that the computational cost of its generation is amortized since computing hashes is computationally inexpensive.

[8] Yacine Challal, Abdelmadjid Bouabdallah, and Yoann Hinard. "RLH: receiver driven layered hash-chaining for multicast data origin authentication". In: *Computer Communications* 28.7 (2005), pp. 726–740.

7.2.3 *Signature Dispersal*

The basic idea behind schemes from the signature dispersal class (such as Tartary, Wang, and Ling's[9]) is that data are divided into fixed-size blocks and each block is signed independently with a digital signature. The signature of a block is split, and each part of the signature is appended to one packet of the block. In this way, only one signature is required for each block, but it can only be verified after all parts of the block have been received.

[9] C. Tartary, Huaxiong Wang, and San Ling. "Authentication of Digital Streams". In: *IEEE Transactions on Information Theory* 57.9 (Sept. 2011), pp. 6285–6303. ISSN: 0018-9448. DOI: 10.1109/TIT.2011.2161960.

7.3 *Designing Fast Authentication Schemes*

A conceptionally entirely different approach to mitigating the computational cost of digital signature schemes is designing fast authentication schemes. We distinguish three different classes that follow the approach of designing fast authentication schemes: OTS, MTS, and unrestricted-time high-speed signing that we suggest.

7.3.1 *OTS Schemes*

Lamport's OTS scheme[10] was published in 1979, although the concept of OTS was first described in 1976 by Diffie and Hellman[11]. As the name suggests, OTS schemes can be used to sign exactly one message securely under a given key pair. This is in stark contrast to conventional signature schemes that can sign a practically unlimited number of messages. OTS schemes are extremely fast, which makes them appealing, nonetheless.

[10] Leslie Lamport. *Constructing digital signatures from a one-way function*. Technical Report CSL-98, SRI International Palo Alto, 1979.

[11] Diffie and Hellman, "New directions in cryptography".

The general idea of Lamport's OTS scheme is that the secret key is the preimage of the public key, i.e., the public key is the hash of the secret key. Another peculiarity of OTS schemes is that parts of the secret key are used as signature and are therefore leaked. If an OTS scheme is used to sign more than one message, the part of the secret key that an adversary may have obtained increases. At some point, the adversary is able to sign arbitrary messages. For this reason, a one-time key pair can only be used to sign a single message securely, which leads to very frequent key generation and key distribution (for the common case where many messages need to be signed). This drawback (in addition to the size of OTSs) induces substantial communication overhead that makes OTS schemes impractical for most real-world group communication applications. Because of their computational efficiency, however, OTS schemes serve as building block for constructing more sophisticated MTS schemes.

To illustrate this, we describe a simple OTS scheme that can sign one bit, i.e., the message to be signed can be either '0' or '1'. Although this simple scheme appears significantly limited at the first glance, it can be easily extended to sign a message of arbitrary length. As any other signature scheme, it consists of three functions: (1) key generation, (2) signing, and (3) verification: (1) The sender generates two distinct key pairs: one key pair for signing the message '0' and another key pair for signing the message '1'. Each key pair consists of a secret key and a public key (which is the hash of the secret key). Both key pairs can be generated very efficiently. After key generation, the sender shares the public keys with the receiver. (2) To sign a message, the sender uses the secret key as signature that corresponds to the message (that is either '0' or '1'). In this way, the sender releases half of the one-time secret key, i.e., the part that corresponds to the message (but not the secret key for the other message). (3) To verify the validity of the signature, the receiver calculates the hash of the signature (i.e., the hash of the secret key) and verifies that it equals the public key of that message. If the signature's hash matches the corresponding public key, the message is accepted as valid. In order to create a valid signature, an adversary would have to find a value that hashes to a specific public key, which is computationally unfeasible (because of the hash function's preimage resistance, see Section 2.1).

To sign a message of arbitrary length, the simple OTS scheme needs to be extended. First, the sender applies the hash-and-sign paradigm so that only a fixed length needs to be signed instead of the entire message. Then, the sender generates a key pair not only for one bit but for each bit of the fixed-size hash and signs each bit of a message individually. Since the signature size of such a scheme is large, many improvements have been suggested but the basic concept of OTS schemes still persists.

7.3.2 *MTS Schemes*

As pointed out in Subsection 7.3.1, OTS schemes are restricted in a way that they can sign only a single message securely. MTS schemes extend OTS schemes to overcome that limitation. Merkle proposed the first MTS scheme in 1979, which eventually got published later in 1989[12]. For this reason, MTS schemes are sometimes also called Merkle signature schemes (or k-time signature schemes).

[12] Ralph C. Merkle. "A certified digital signature". In: *Advances in Cryptology—CRYPTO*. Springer, 1989, pp. 218–238.

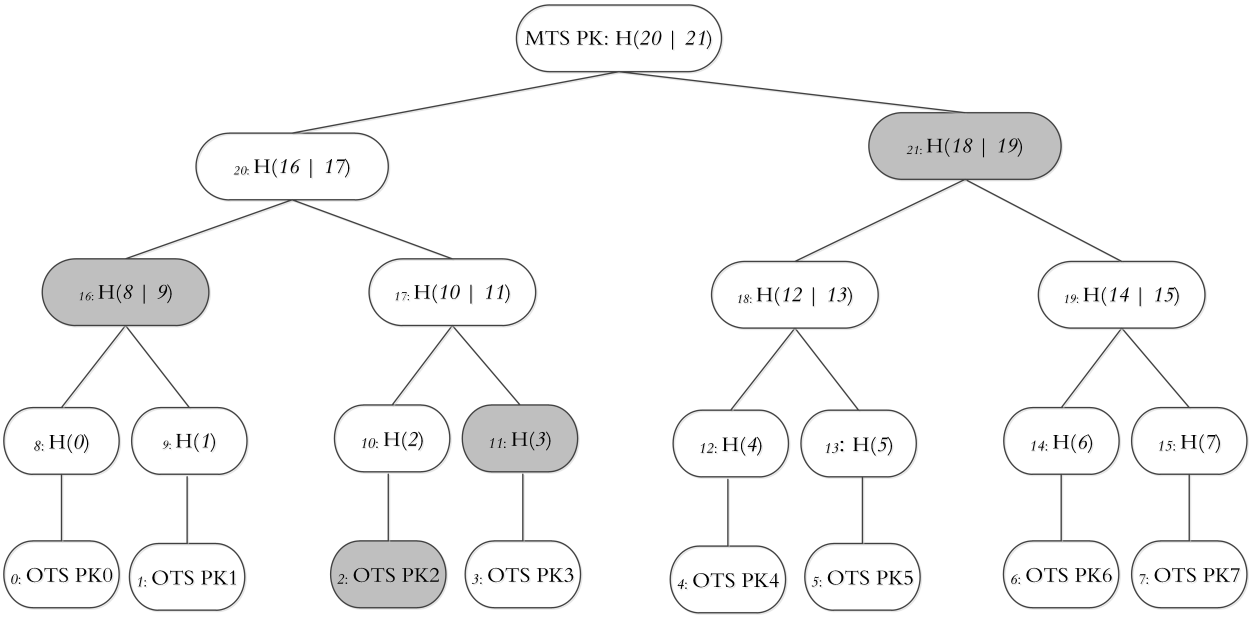

Figure 7.2: Merkle tree construction. Numbers in the graph are just labels for the nodes. *PK* is short for public key.

The basic idea of MTS schemes is that they combine multiple OTS key pairs into a single MTS key pair. An MTS scheme consists of a fixed number of OTS key pairs and an indexing function that selects which OTS key is to be used for a particular message. For this reason, an MTS scheme can sign a fixed number of messages under one MTS key pair. To combine multiple OTS keys into one MTS key, a one-way chain or a binary authentication tree (a Merkle tree) can be employed. We consider the following as an exemplary MTS scheme using Merkle trees (as one-way chains have already been described in Section 2.2.).

Each OTS public key is stored in one leaf of the authentication tree. Each parent node contains the hash of the concatenation of its two child nodes. The root of the tree on the top represents the overall MTS public key, as shown in Figure 7.2. When signing a message, the indexing function defines which OTS secret key is used as signature. In this example, the OTS secret key "2" is used to sign the message. In addition to the OTS secret key, a sequence from the leaf to the root of the tree is part of the MTS signature. The OTS secret key together with this sequence build the MTS signature for the message. This sequence from the leaf to the root of the tree is called *authentication path* (depicted as gray nodes in Fig. 7.2). The authentication path needs to contain the parts of the tree that (together with the OTS secret key) lead to the root of the tree, which is the MTS public key.

So with the hash of the OTS secret key "2" (i.e., "10") plus its neighbor ("11"), their parent ("17") can be calculated. Together with the parent's neighbor ("16"), their parent ("20") can be calculated. Eventually, the root of the tree (i.e., the MTS public key) can be calculated. For successful verification of a message, the OTS secret key selected by the indexing function must match the corresponding part of the signature. Furthermore, the OTS secret key and the authentication path must lead to the root of the tree; i.e., the MTS public key. If both checks succeed, the message is accepted; otherwise, it is rejected.

7.3.3 Unrestricted-Time High-Speed Signing

In this subsection, we introduce a new class of data origin authentication schemes: unrestricted-time high-speed signing, which employs novel high-speed signature schemes. An implicit assumption from schemes in other classes is that digital signature schemes are computationally too expensive to sign many consecutive packets individually. This assumption, however, only holds for conventional but not necessarily for novel high-performance signature schemes (as to be assessed in Chapter 9). In recent years, novel signature schemes have been proposed that offer previously unrivaled performance. Employing such novel high-performance signature schemes can mitigate the negative performance impact perceived from conventional schemes. For this reason, we argue to sign every packet independently in unrestricted-time high-speed signing despite the common assumption that it would be impractical due to the computationally cost.

8

Evaluation of Data Origin Authentication Schemes

This chapter has been published in part in:

Robert Annessi, Tanja Zseby, and Joachim Fabini. "A new Direction for Research on Data Origin Authentication in Group Communication". In: *International Conference on Cryptology and Network Security (CANS)*. Springer, 2017. DOI: 10.1007/978-3-030-02641-7_26. Referred to as "[I]". Adapted with permission from Springer Nature Customer Service Centre GmbH. © Springer Nature 2018.

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Robert Annessi, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252. Referred to as "[IV]". © ACM 2018.

Data origin authentication schemes for group communication have been researched for more than twenty-five years, and many ideas were proposed to solve this challenging problem. In this chapter, we evaluate data origin authentication schemes for their suitability to secure group communication generally and assess to which degree existing authentication schemes fulfill the requirements identified in Chapter 6 (i.e., computational efficiency, scalability, low communication overhead, immediate signing, immediate verification, collusion resistance, information-theoretical security, robustness to packet loss, independence of clock synchronization, secure channel only initially required, support for a wide range of applications, and resilience against DoS attacks). The evaluation shows that each class of schemes comprises a trade-off from a specific point of view. None of the proposed schemes, however, satisfies all constraints and requirements of applications so that naming a single superior scheme seems non-trivial[1]. We argue that the unrestricted-time high-speed signing class does not require any of those trade-offs. We use the classification of data origin authentication schemes introduced in Chap-

[1] Steinwandt and Villányi, "A One-time Signature Using Run-length Encoding".

ter 7: extending symmetric schemes for data origin authentication with the secret-information asymmetry class, reducing the cost of conventional signature schemes with the deferred signing, signature propagation, and signature dispersal classes, and designing fast authentication schemes with the OTS, MTS, and unrestricted-time high-speed signing classes.

8.1 Extending Symmetric Schemes for Data Origin Authentication

8.1.1 Secret-Information Asymmetry

The secret-information asymmetry class is unique in a sense that most schemes are information-theoretically secure[2], which means that they do not provide enough information to enable attacks in the first place. In this way, schemes from the secret-information asymmetry class protect against adversaries with potentially unbounded computational power. They are furthermore resistant to packet loss (as each packet is signed individually), independent of clock synchronization, and provide immediate signing and verification of packets. However, secret-information asymmetry schemes are prone to collusion of receivers, where dishonest or compromised receivers collaborate in order to reconstruct the sender's entire set of keys to impersonate the sender. Communication overhead increases with the number of receivers so that the scalability of schemes from the secret-information class is limited. Furthermore, secret-information asymmetry schemes require distributing new keys individually to each receiver frequently[3] for which a secure channel between sender and receiver is required permanently throughout the communication. The need to distribute new keys often, make information-theoretically secure schemes impractical for most real-world applications.

[2] An authentication schemes is considered information-theoretical secure (or unconditionally secure) when an attacker without knowledge of the key cannot create a valid signature even if the attacker has unbounded computational and time resources.

[3] I.e., for each message and for each change in group membership.

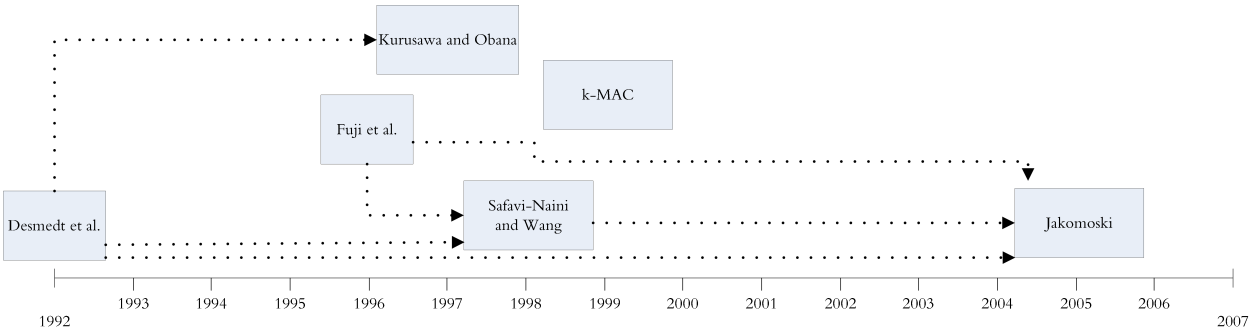

Figure 8.1: Timeline of secret-information asymmetry schemes.

In the following, we show the historical evolution of secret-information asymmetry schemes.

Desmedt In 1992, Desmedt, Frankel, and Yung were the first to propose a data origin authentication scheme for group communication[4]. Their scheme is similar to Shamir's secet sharing[5] as both employ polynomial interpolation to achieve security. In Desmedt, Frankel, and Yung's data origin authentication scheme, the key is separated into shares and each receiver gets to know a different share. The scheme is constructed such that $t + 1$ shares can reconstruct the key but $\leq t$ shares cannot. The sender generates the authenticator polynomial using Lagrange interpolation, and each receiver verifies it using

[4] Desmedt, Frankel, and Yung, "Multi-receiver/multi-sender network security".

[5] Adi Shamir. "How to Share a Secret". In: *Commun. ACM* 22.11 (Nov. 1979), pp. 612–613.

its share. The scheme influenced other schemes from the secret-information asymmetry class (see Fig. 8.1).

Kurusawa and Obana In 1997, Kurosawa and Obana determined lower bounds for the key sizes of authentication schemes from the secret-information asymmetry class as well as the lower bounds for impersonation and substitution attacks[6,7]. Furthermore, they showed that the scheme by Desmedt, Frankel, and Yung[8] satisfies these bounds.

Safavi-Naini and Wang Based on Cover-Free Family (CFF), Safavi-Naini and Wang[9,10] generalized the Desmedt, Frankel, and Yung scheme in such a way that each polynomial can be used to authenticate multiple messages (instead of just one). A scheme proposed by Fujii, Kachen, and Kurosawa[11] can be considered a special case of Safavi-Naini and Wang.

k-MAC The basic idea in Canetti et al.'s k-MAC[12] is that the sender knows several keys, and a subset of these keys is shared with each receiver such that no group of up to k receivers should know the set of keys known by any receiver (that is not part of that group). The number of keys required overall is directly related to the maximum size (k) of the colluding group. The more potential colluders (i.e., the larger k), the more keys are needed. The sender authenticates a message by computing MACs with each key. Then, all MACs are appended to the message. Upon reception of a message, each receiver can verify the MACs it knows the keys for but cannot create valid authentication information on behalf of the sender (as the other keys are unknown).

If more than k receivers collude, however, the colluders can reconstruct a set of keys such that the security of the scheme breaks. Therefore, k-MAC is only suitable for applications where the number of expected colluders is known and small. This is a problem in group communication, however, because group communication is supposed to handle very large groups, and the size of the group and the number of colluders is likely correlated. For this reason, k-MAC is not suitable to a wide range of applications. Nevertheless, since the k-MAC approach employs computationally efficient MACs, signature generation and verification are fast. To reduce the communication overhead caused by the number of MACs appended to each message, Canetti et al. suggest one-bit MACs so that the overall communication overhead per message is reduced.

Jakimoski Jakimoski aimed to design data origin authentication schemes that are unconditionally secure and packet-loss resistant at the same time by means of erasure-tolerant authentication codes[13]. To this end, Jakimoski proposes to employ Reed-Solomon codes to construct erasure-tolerant authentication codes. However, adversaries are not allowed to inject packets[14] so that this scheme is only suitable to very specific scenarios.

8.2 Reducing the Cost of Conventional Signature Schemes

8.2.1 Deferred Signing

Schemes from the deferred signing class aim to reduce the computational cost of conventional signature schemes by employing a two-phase approach: (1) a

[6] Kaoru Kurosawa and Satoshi Obana. "Characterization of (k, n) multi-receiver authentication". In: *Information Security and Privacy*. Lecture Notes in Computer Science 1270. Springer, July 7, 1997, pp. 204–215. ISBN: 978-3-540-69237-9.

[7] Satoshi Obana and Kaoru Kurosawa. "Bounds and Combinatorial Structure of (k,n) Multi-Receiver A-Codes". In: *Designs, Codes and Cryptography* 22.1 (Jan. 2001), pp. 47–63. ISSN: 0925-1022, 1573-7586. DOI: 10.1023/A:1008351225940.

[8] Desmedt, Frankel, and Yung, "Multi-receiver/multi-sender network security".

[9] R. Safavi-Naini and H. Wang. "New results on multi-receiver authentication codes". In: *Advances in Cryptology — (EUROCRYPT)*. Lecture Notes in Computer Science 1403. Springer, May 31, 1998, pp. 527–541. ISBN: 978-3-540-69795-4.

[10] R. Safavi-Naini and H. Wang. "Multireceiver Authentication Codes: Models, Bounds, Constructions, and Extensions". In: *Information and Computation* 151.1–2 (May 25, 1999), pp. 148–172. ISSN: 0890-5401. DOI: 10.1006/inco.1998.2769.

[11] Hiroshi Fujii, Wattanawong Kachen, and Kaoru Kurosawa. "Combinatorial bounds and design of broadcast authentication". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 79.4 (1996), pp. 502–506.

[12] Canetti et al., "Multicast security".

[13] Goce Jakimoski. "Unconditionally Secure Information Authentication in Presence of Erasures". In: *Cryptography and Coding*. Lecture Notes in Computer Science 3796. Springer, Dec. 19, 2005, pp. 304–321. ISBN: 978-3-540-32418-8.

[14] Tartary, Wang, and Ling, "Authentication of Digital Streams".

slow offline phase, in which one-time keys are generated and each one-time key is signed with a conventional signature scheme in order to ensure that the one-time key indeed originates from the claimed sender. In the fast online phase (2) the one-time keys are used with a computationally efficient OTS scheme to sign messages. The OTS as well as the (certified) signature of the one-time key are attached to the message. In this way, the computationally intensive tasks (i.e., precomputing the one-time keys and signing each of them with a conventional signature scheme) do not interfere with online communications. The computational effort required in the offline part is substantial, however, and the communication overhead is significant because of the size of the OTSs.

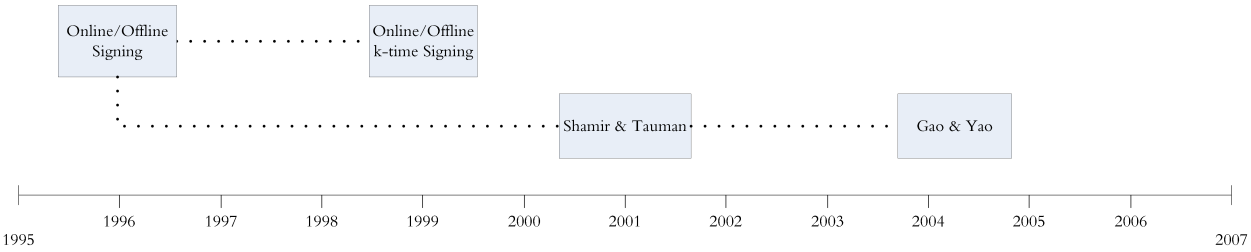

Figure 8.2: Timeline of deferred signing schemes.

Offline/Online Signing The first scheme from the deferred signing class, offline/online signing[15], was published by Even, Goldreich, and Micali in 1996. Each one-time public-key is signed with a conventional signature to certify that the one-time public key originates from the claimed sender. Since one-time public keys can be signed offline[16], it does not negatively impact online message signing performance. During the online phase, however, receivers have to verify the signature for the one-time public key from a conventional signature scheme in addition to the OTS on the message. The performance of OTSs is, therefore, impaired by the use of conventional signatures, which is a serious performance drawback. Furthermore, the size of OTSs is very large, which implies significant communication overhead. The offline/online signing scheme is resistant to collusion of receivers, tolerates packet loss (because packets are signed individually), is independent of clock synchronization, and packets can be signed and authenticated immediately.

[15] Even, Goldreich, and Micali, "On-line/off-line digital signatures".

[16] And the signed one-time public keys are sent to the receivers before the communication.

Offline/Online k-time Signing In 1999, Rohatgi improved the offline/online signing scheme by creating the offline/online k-time signing scheme[17], in which k-time keys are employed instead of one-time keys. In this way, the computationally most expensive operation, signing keys with a conventional digital signature scheme, can be amortized over multiple packets as the sender uses each key to sign k messages instead of just one. Therefore, less conventional signatures are required overall. Since verification of the conventional signature is the most expensive (online) operation, computational efficiency is improved. Packet loss resistance suffers from this approach, however, because if the packet containing the k-time keys' signature is dropped, none of the k messages can be verified. If that signature packet is sent more often, communication overhead increases, and the problem is just mitigated (but not

[17] Pankaj Rohatgi. "A Compact and Fast Hybrid Signature Scheme for Multicast Packet Authentication". In: *ACM Conference on Computer and Communications Security*. CCS. USA: ACM, 1999, pp. 93–100. ISBN: 1-58113-148-8. DOI: 10.1145/319709.319722.

solved). So basically there is a trade-off between computational efficiency and packet loss resistance in offline/online k-time signing. Since one disadvantage of the offline/online k-time signature scheme is its communication overhead of around 1 kB per packet (because of the OTS's size), Rohatgi provided a method to reduce the communication overhead. This improvement in communication overhead negatively affects computational efficiency, however. Furthermore, the offline/online k-time signing scheme requires frequent key generation and distribution throughout the communication, since the keys can be used to sign only k messages. For this purpose, a reliable channel from the sender to the receivers is required periodically, in order to distribute new k-time keys.

Offline/Online Signing Improvements In 2001, Shamir and Tauman reduced the size of offline/online signatures using trapdoor hash functions[18]. Their approach was further improved by Gao and Yao in 2005 who additionally reduced the computational cost (but require the length of the data to be known in advance)[19]. Although the differed signing class did not see any further developments since then, it is still important as a general concept.

[18] Adi Shamir and Yael Tauman. "Improved online/offline signature schemes". In: *Advances in Cryptology*. Springer, 2001, pp. 355–367.

[19] Chong-zhi Gao and Zheng-an Yao. "How to Authenticate Real Time Streams Using Improved Online/Offline Signatures". In: *Cryptology and Network Security*. Lecture Notes in Computer Science 3810. Springer, 2005, pp. 134–146. ISBN: 978-3-540-32298-6.

8.2.2 Signature Propagation

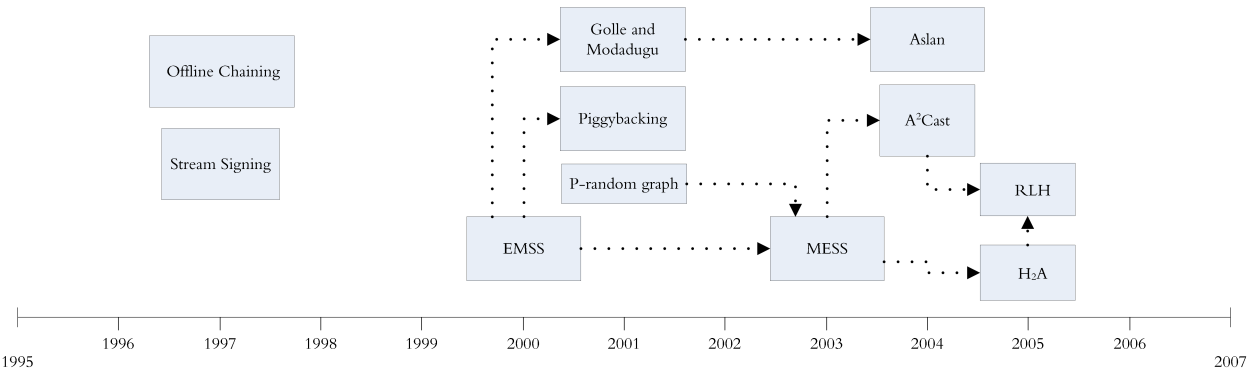

Figure 8.3: Timeline of signature propagation schemes.

With schemes from the signature propagation class, a signature propagates through all packets so that the computational cost of its generation is amortized (as hash operations are computationally inexpensive). Signature propagation schemes, however, require packets to be buffered at the sender or at the receiver before they can be signed or their signature be verified, respectively. Such buffering introduces additional delay that may be intolerable to specific applications, such as real-time applications. Receiver-side buffering additionally increases the risk for DoS attacks as buffers may be filled with bogus packets by an attacker with access to the network. Furthermore, signature propagation schemes rely on the successful reception of signature packets and are therefore not entirely resistant to packet loss. Since packets may be lost during transmission, some solutions introduce redundancy so that authentication information can be recovered. The higher the robustness to packet loss, the larger also the communication overhead. Signature propagation schemes that aim to provide resistance against packet loss try to minimize the communication overhead while maximizing packet loss resistance. Fig. 8.3 depicts

the timeline and relation of schemes from the signature propagation class, and Fig. 8.4 depicts the principle operation of signature propagation schemes.

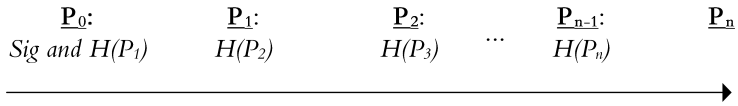

Figure 8.4: Basic principle of signature propagation schemes. The signature included in the first packet propagates through all packets, as each packet contains the hash of the subsequent packet (offline chaining).

Offline Chaining The first scheme in the signature propagation class was proposed in 1997 by Gennaro and Rohatgi[20,21]. Actually, they proposed two schemes: one that signs data offline (offline chaining) and assumes the data to be known to the sender upfront, and another scheme where there is no such restriction (stream signing).

In the offline chaining scheme, only the first packet is signed with a conventional signature scheme. Each packet includes an association (i.e., a hash) to the next packet which in turn includes the hash of the subsequent packet, and so on. For this reason, the subsequent packet must be known in advance, but the computational cost of conventional digital signatures are amortized over all messages as only the first packet needs to be signed while all other packets need only a computationally efficient hash function such that authentication information is reduced to (roughly) one hash per packet. The offline chaining approach by Gennaro and Rohatgi works only for a finite number of messages, which need to be known to the sender in advance. For this reason, the scheme is not suitable to a wide range of applications. However, the scheme could be modified such that requiring knowledge of the entire stream in advance is replaced by sender-side buffering. Nevertheless, the scheme cannot provide resistance to packet loss: if a single packet is lost, the authentication chain breaks such that no signature of subsequent packets can be recovered. The sender has to furthermore buffer one packet (that contains the hash of the next), which introduces additional delay.

Stream Signing In their stream signing scheme, Gennaro and Rohatgi again employ a conventional signature scheme. The stream signing scheme improves upon the offline chaining scheme in a way that one-time keys are employed so that the sender is not required to buffer packets (to include the hash of the subsequent packet). In stream signing, the (conventional) signature scheme is used to sign a one-time key that is used to sign the subsequent packet. Authentication information therefore consists of a OTS plus a signed one-time public key per message that is then used for signing the subsequent packet. The one-time public key can then be used by the receiver to verify the authenticity of the message, which in turn contains another one-time key for the subsequent message. In this way, there is no need for the sender to buffer one message until the next is known (or to even know all messages in advance). In contrast to the offline chaining scheme, Gennaro and Rohatgi's stream signing works for infinite number of messages.

Another benefit is the computational efficiency of OTS schemes, which allow signing messages at a high rate. However, the stream signing scheme is computationally less efficient overall than the offline chaining scheme because each one-time key needs to be signed with a conventional signature during

[20] Rosario Gennaro and Pankaj Rohatgi. "How to sign digital streams". In: vol. Advances in Cryptology. Springer, 1997, pp. 180–197.

[21] Rosario Gennaro and Pankaj Rohatgi. "How to Sign Digital Streams". In: *Information and Computation* 165.1 (2001), pp. 100–116. ISSN: 0890-5401. DOI: 10.1006/inco.2000.2916.

the offline phase and receivers need to verify both a one-time signature and a conventional signature for each packet. The scheme has significant communication overhead (due to the size of OTSs) but is resistant to collusion of receivers and independent of clock synchronization.

EMSS The Efficient Multi-chained Stream Signature (EMSS)[22] scheme by Perrig et al. was the first scheme to introduce redundancy in hash chaining and inspired many other schemes. In EMSS, the hash of a packet is appended deterministically to multiple other packets. A signature packet signs the hash of a few packets (that contain hashes of previous packets) is sent from time to time in order to provide non-repudiation. Receivers, therefore, need to buffer packets until they receive a signature packet in order to verify their authenticity. To reduce the buffering pressure on receivers, EMSS divides a stream into fixed-size blocks. A packet remains verifiable as long as a hash-link path through packets remains to a signature packet. When increasing the number of hashes embedded per packet, the probability of successfully authenticating (valid) packets increases. This also means, however, that high verification rates come with increased communication overhead.

[22] Adrian Perrig et al. "Efficient authentication and signing of multicast streams over lossy channels". In: *IEEE Symposium on Security and Privacy (S&P)*. 2000, pp. 56–73.

RLH RLH[23] can be considered a variant of EMSS, which adds data to the communication that contain hashes of other packets to increase resistance to packet loss. RLH aims to maximize authentication probability while minimizing communication overhead. RLH uses two constructions for determining the position of the added packets (containing the hashes) in the data stream: mix deterministic and random hash distribution. RLH does not require a feedback channel between sender and receiver; instead, each receiver chooses which hash chain to use according to its own packet verification probability. While packet loss resistance is improved by combining both constructions, RLH is not resistant against attacks by an adversary who intentionally drops packets containing the hashes. Furthermore, computational efficiency and communication overhead are slightly worse than in other schemes from the signature propagation class, and receivers have to buffer messages before authentication.

[23] Challal, Bouabdallah, and Hinard, "RLH: receiver driven layered hash-chaining for multicast data origin authentication".

Augmented Chain Technique In 2001, Golle and Modadugu proposed two schemes[24] that aim to resist bursty packet loss by inserting hashes in strategic locations. Their first scheme is a special case of EMSS (where the sender is not supposed to buffer packets) that can resist packet loss bursts of length $a-1$, where a is the maximum distance of a hash to the corresponding packet. The basic idea is that the length of the maximal packet loss burst is known $(a-1)$ and the hash of a packet is not only added to the subsequent packet but also to the $a-1^{th}$ packet thereafter. Their second scheme can resist bursts up to a length of $p \cdot (a-1)$ where p is the number of packets the sender can buffer. For the second scheme, it is assumed that the sender can buffer packets, in addition to the receivers.

[24] Philippe Golle and Nagendra Modadugu. "Authenticating Streamed Data in the Presence of Random Packet Loss." In: *NDSS*. 2001, pp. 13–22.

The p-random Authentication Scheme The p-random authentication scheme[25] is a probabilistic authentication scheme based on hash-chaining that is designed to resist random packet loss. Senders are assumed to know the data in advance

[25] Sara Miner and Jessica Staddon. "Graph-based authentication of digital streams". In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2001, pp. 232–246.

such that the signature packet can be sent first, which includes the hashes of later packets. For this reason, the *p*-random scheme is not suitable to a wide range of applications, but receivers can immediately verify the authenticity of packets.

Piggybacking To resist bursty packet loss, Miner and Staddon designed another authentication scheme called Piggybacking[26]. In Piggybacking, packets are associated with different priority classes such that more important packets have increased burst tolerance. The scheme requires buffering at both sender and receiver sides.

MESS In 2003, Perrig and Tygar[27] published an improved variant of EMSS, which they named MESS. While the positions of hash packets in a stream is strictly deterministic in EMSS, it is randomized in MESS. The authors showed that such randomization provides probabilistic robustness to packet loss (similar to the p-random authentication scheme).

A^2Cast and H2A Challal, Bettahar, and Bouabdallah proposed two similar extensions to EMSS: A^2Cast[28] and Hybrid Hash-chaining scheme for adaptive source Authentication (H2A)[29]. Both schemes rely on a feedback channel between sender and receiver in order to adapt the distribution of hash packets to mitigate different packet loss scenarios. These schemes involve a mix of deterministic and random hash distributions to reduce the overhead and increase authentication probability. The same authors also published other schemes[30,31].

Aslan In the scheme[32] proposed by Aslan in 2004, the hash chain is constructed in reverse order: the signature packet is sent first so that received packets can be authenticated immediately. This is in contrast to many other schemes from the signature propagation class (but similar to offline chaining and the *p*-random authentication scheme).

[26] Miner and Staddon, "Graph-based authentication of digital streams".

[27] Adrian Perrig and J. D. Tygar. *Secure Broadcast Communication*. Springer, 2003. ISBN: 978-1-4615-0229-6.

[28] Y. Challal, H. Bettahar, and A. Bouabdallah. "A^2cast: an Adaptive source Authentication protocol for multiCAST streams". In: *International Symposium on Computers and Communications*. June 2004, 363–368 Vol.1. DOI: 10.1109/ISCC.2004.1358431.

[29] Yacine Challal, Abdelmadjid Bouabdallah, and Hatem Bettahar. "H_2A: Hybrid Hash-chaining scheme for Adaptive multicast source authentication of media-streaming". In: *Computers & Security* 24.1 (Feb. 2005), pp. 57–68. ISSN: 0167-4048. DOI: 10.1016/j.cose.2004.06.012.

[30] Yacine Challal, Hatem Bettahar, and Abdelmadjid Bouabdallah. "Hybrid and Adaptive Hash-Chaining Scheme for Data-Streaming Source Authentication". In: *High Speed Networks and Multimedia Communications*. Lecture Notes in Computer Science 3079. Springer, 2004, pp. 1056–1067. ISBN: 978-3-540-25969-5.

[31] Yacine Challal and Abdelmadjid Bouabdallah. "Authenticast: a source authentication protocol for multicast flows and streams". In: *International Conference on Information Security*. 2005, pp. 175–178.

[32] Heba K. Aslan. "A hybrid scheme for multicast authentication over lossy networks". In: *Computers & Security* 23.8 (Dec. 2004), pp. 705–713. ISSN: 01674048. DOI: 10.1016/j.cose.2004.06.010.

8.2.3 Signature Dispersal

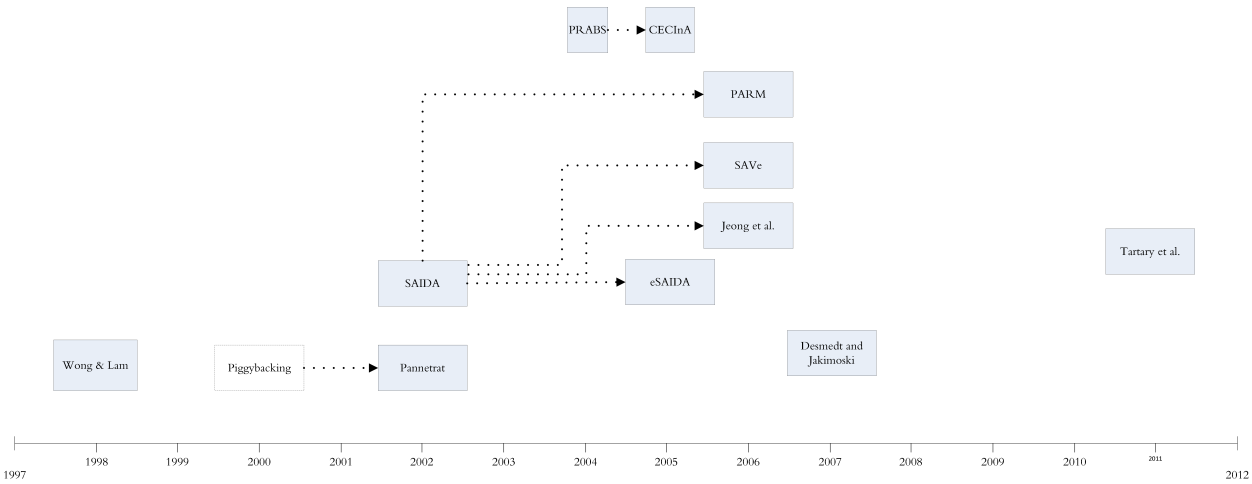

Figure 8.5: Timeline of signature dispersal schemes.

Signature dispersal is the third class of data origin authentication schemes that aim to reduce the cost of conventional signature schemes. In the signature dispersal class, the idea is to split data into fixed-sized blocks and to be robust to packet loss, which was learned to be essential from schemes in the signature propagation class.

But in signature dispersal schemes, the signature of each block is split into smaller parts to increase robustness to packet loss, and additional information is added to each packet that helps the receivers to reconstruct the signature even if some packets were lost. Each part of the signature (plus the additional information) is then appended to one packet within the block. In this way, schemes from the signature dispersal class improve packet loss resistance compared to signature propagation schemes that rely on the reception of signature packets entirely. Computational efficiency is reduced, however, and receivers need to wait for the whole block before they can verify the authenticity of that block (and of the included packets). Fig. 8.5 on the previous page depicts the timeline and relation of schemes from the signature dispersal class, and Fig. 8.6 depicts the principle operation of signature dispersal schemes.

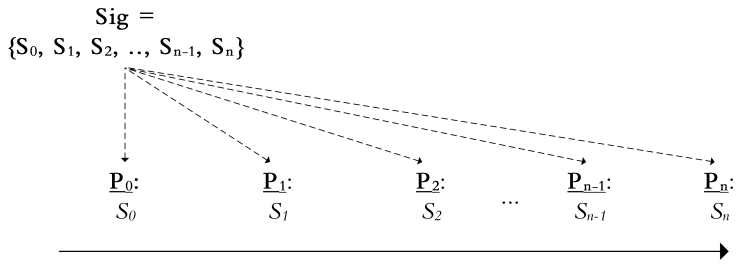

Figure 8.6: Basic principle of signature dispersal schemes: the signature dispersed over all packets.

Wong-Lam In 1998 and 1999 Wong and Lam proposed the first data origin authentication scheme[33,34] from the signature dispersal class. A stream of data is separated into time intervals, and for each time interval a Merkle tree is constructed from all packets collected during that interval. Each packet contains an authentication path from the leaf to the root of the tree (similar to some MTS schemes as explained in Section 7.3.2). The root of the tree contains a signature of the time interval from a conventional signature scheme. In this way, a valid path from a leaf to the root enables authentication using hash functions only. The approach proposed by Wong and Lam is perfectly resistant to packet loss as every packet carries its own authentication information. The sender cannot immediately sign packets, however, the number of packets that can be signed (during a time interval) is limited, and communication overhead is significant.

SAIDA Park, Chong, and Siegel propose to use the Information Dispersal Algorithm (IDA)[35] to disperse a signature over multiple packets. For this reason, the scheme is called Signature Amortization using IDA (SAIDA)[36,37]. SAIDA trades off resistance to packet loss with the size of the authentication information, i.e., communication overhead. Nevertheless, it introduces significant computational cost due to the use of IDA. It is resistant to collusion attacks of receivers and independent of clock synchronization. However, it requires both sender and receivers to buffer messages before signing and veri-

[33] Chung Kei Wong and S.S. Lam. "Digital signatures for flows and multicasts". In: *International Conference on Network Protocols*. Oct. 1998, pp. 198–209. DOI: 10.1109/ICNP.1998.723740.

[34] Chung Kei Wong and S.S. Lam. "Digital signatures for flows and multicasts". In: *IEEE/ACM Transactions on Networking* 7.4 (Aug. 1999), pp. 502–513. ISSN: 1063-6692. DOI: 10.1109/90.793005.

[35] Michael O. Rabin. "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance". In: *J. ACM* 36.2 (Apr. 1989), pp. 335–348. ISSN: 0004-5411. DOI: 10.1145/62044.62050.

[36] Jung Min Park, Edwin KP Chong, and Howard Jay Siegel. "Efficient multicast packet authentication using signature amortization". In: *IEEE Symposium on Security and Privacy*. IEEE, 2002, pp. 227–240.

[37] Jung Min Park, Edwin KP Chong, and Howard Jay Siegel. "Efficient multicast stream authentication using erasure codes". In: *ACM Transactions on Information and System Security (TISSEC)* 6.2 (2003), pp. 258–285.

fication, respectively. Furthermore, the scheme is only resistant to packet loss for a specific, predefined packet loss ratio.

SAIDA influenced many researchers who proposed improvements. Park and Cho, for example, designed enhanced SAIDA (eSAIDA)[38], which improves computational efficiency as well as authentication probability by modifying the computation slightly. Pannetrat and Molva proposed a similar approach[39,40] to SAIDA, where they use an erasure code instead of IDA to disperse authentication information. Jeong, Park, and Cho proposed an authentication scheme[41] similar to SAIDA as well but with error correction codes instead of IDA. Lin, Shieh, and Lin proposed Pollution Attack Resistant Multicast authentication (PARM)[42], an improvement to SAIDA that has smaller communication overhead and is computationally more efficient. Ueda et al. proposed a scheme called Stream Authentication scheme for Videos (SAVe)[43,44] that allows to configure the redundancy added to each packet according to its importance (i.e., more redundancy is added for more important packets). SAVe improves authentication probability of packets in case of packet loss.

Desmedt and Jakimoski In 2007, Desmedt and Jakimoski proposed to use CFFs both to reduce redundancy in authentication information[45] and to provide packet loss resistance.

Tartary et al. To counter packet injection attacks, Tartary, Wang, and Ling[46] proposed to use list recoverable codes which allows receivers to detect injected packets efficiently. Furthermore, the authors show that various other schemes from the signature dispersal class can be treated as special cases of their scheme. The scheme expects the fraction of received packets to be above a predefined threshold, and the fraction of injected packets below some other predefined threshold. As long as these assumptions hold, the scheme can recover lost authentication information entirely. The scheme is independent of clock synchronization and resistant to collusion of receivers. The communication overhead and computational efficiency are acceptable, but packets need to be buffered before they can be signed and before they can be authenticated.

8.3 Designing Fast Authentication Schemes

8.3.1 OTS Schemes

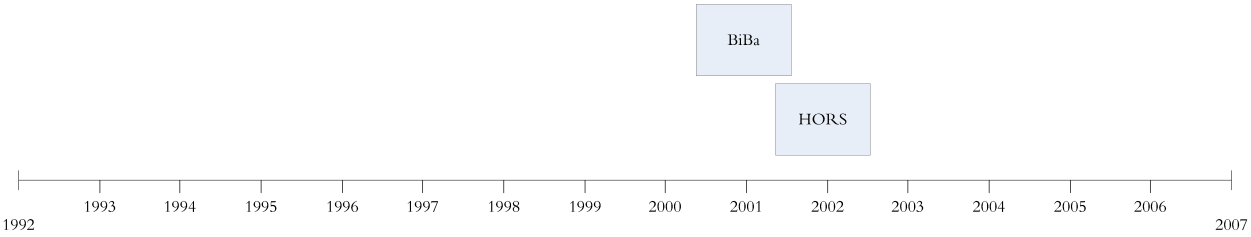

Figure 8.7: Timeline of OTS schemes.

In Section 7.3.1, we have shown the Lamport's OTS as basic example of an OTS scheme. Since 1979 other OTS schemes were proposed. Two of them

[38] Yongsu Park and Yookun Cho. "The eSAIDA Stream Authentication Scheme". In: *Computational Science and Its Applications – ICCSA*. Lecture Notes in Computer Science 3046. Springer, May 14, 2004, pp. 799–807. ISBN: 978-3-540-24768-5.

[39] Alain Pannetrat and Réfik Molva. "Authenticating real time packet streams and multicasts". In: *International Symposium on Computers and Communications, ISCC*. IEEE, 2002, pp. 490–495.

[40] Alain Pannetrat and Refik Molva. "Efficient Multicast Packet Authentication." In: *NDSS*. 2003.

[41] JaeYong Jeong, Yongsu Park, and Yookun Cho. "Efficient DoS Resistant Multicast Authentication Schemes". In: *Computational Science and Its Applications – ICCSA*. Lecture Notes in Computer Science 3481. Springer, 2005, pp. 353–362. ISBN: 978-3-540-32044-9.

[42] Ya-Jeng Lin, Shiuhpyng Shieh, and Warren W. Lin. "Lightweight, Pollution-attack Resistant Multicast Authentication Scheme". In: *ACM Symposium on Information, Computer and Communications Security*. ASIACCS. USA, 2006, pp. 148–156. ISBN: 978-1-59593-272-3. DOI: 10.1145/1128817.1128840.

[43] S. Ueda et al. "Authenticating Video Streams". In: *International Conference on Advanced Information Networking and Applications, (AINA)*. Apr. 2006, pp. 863–868. DOI: 10.1109/AINA.2006.107.

[44] Shintaro Ueda et al. "A Real-Time Stream Authentication Scheme for Video Streams". In: *Information and Media Technologies* 1.2 (2006), pp. 1014–1024.

[45] Yvo Desmedt and Goce Jakimoski. "Non-degrading Erasure-Tolerant Information Authentication with an Application to Multicast Stream Authentication over Lossy Channels". In: *Topics in Cryptology – (CT-RSA)*. Lecture Notes in Computer Science 4377. Springer, Feb. 5, 2007, pp. 324–338. ISBN: 978-3-540-69328-4.

[46] Tartary, Wang, and Ling, "Authentication of Digital Streams".

are highly relevant because they influenced many data origin authentication schemes for group communication: Bins and Balls (BiBa)[47] and HORS[48]. For both, the communication overhead is significant because of the OTSs' size. HORS signatures are shorter, however, and verification times faster. OTS schemes can only sign a single message securely by design. For this reason, many one-time keys need to be generated and distributed, which makes OTS schemes rather unpractical for most real-world applications. Nevertheless, OTS schemes serve as essential building block of many MTS schemes.

[47] Adrian Perrig. "The BiBa One-time Signature and Broadcast Authentication Protocol". In: *ACM Conference on Computer and Communications Security*. CCS. New York, USA: ACM, 2001, pp. 28–37. ISBN: 1-58113-385-5. DOI: 10.1145/501983.501988.

[48] Leonid Reyzin and Natan Reyzin. "Better than BiBa: Short one-time signatures with fast signing and verifying". In: *Information Security and Privacy*. Springer, 2002, pp. 144–153.

BiBa OTS BiBa is short for Bins and Balls (BiBa). The BiBa OTS scheme[49] requires a set of random numbers (balls) and a hash function. For signing, a hash is computed on the message and each ball. The number of balls and the hash's size are chosen such that at least two balls fall into the same bin with high probability. The balls that fell into the same bin are used as signature. For verification, it needs to be checked that the hashes of the two balls (and the message) are actually identical. In this way, computational efficiency is achieved as hash functions are very efficient.

[49] Perrig, "The BiBa One-time Signature and Broadcast Authentication Protocol".

HORS Another OTS scheme based on hash functions was proposed by Reyzin and Reyzin in 2002. Their OTS scheme was named HORS[50] and has inspired many variants. HORS uses random numbers and a hash function as well (like BiBa), but signature generation in HORS requires just a single hash function computation. For this reason, signature generation is faster, while verification is as fast in HORS as in BiBa. Furthermore, key and signature sizes are slightly improved in HORS.

[50] Reyzin and Reyzin, "Better than BiBa".

8.3.2 *MTS Schemes*

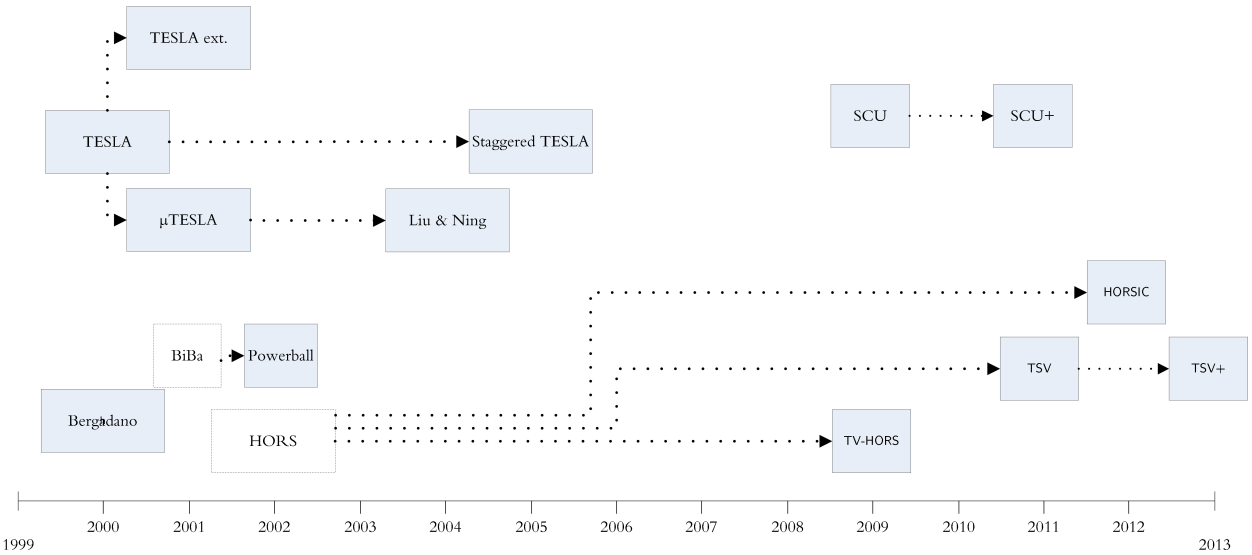

Figure 8.8: Timeline of MTS schemes.

MTS schemes improve upon OTS schemes such that they sign multiple messages securely. MTS have much lower computation requirements than conventional signature schemes (but can sign only a fixed number of messages). Since the number of messages that can be signed under one MTS key pair has

to be fixed in advance, a new key pair needs to be generated and distributed as soon as this limit is reached. However, a secure out-of-band channel for key distribution may not be available to all applications. MTS schemes produce longer signatures at lower security levels compared to conventional signature schemes. Packet loss resistance is provided as packets are signed independently of each other.

TESLA One particular way to drastically reduce the computational effort required for signature generation and verification is using time as a one-way function. In TESLA[51,52,53], key asymmetry is achieved through a common notion of time—the secret and public keys have identical values and are separated only through time. Communication is split into time intervals, and each key of the one-way chain is used to sign packets during one interval. The key used to sign packets is then disclosed in the subsequent time interval. While the key is secret, the sender uses it to sign messages. Receivers buffer packets and can verify their authenticity after the key has been disclosed. A common notion of time guarantees receivers accept signatures under a particular key only during the signing interval but not after the key has been disclosed. For this reason, synchronization of sender and receiver clocks becomes a security requirement so that receivers know when to consider a key as valid (or invalid). Once the key is disclosed, the sender switches to a new secret key in order to sign new messages. Keys are associated using a one-way chain in TESLA so that only the initial key needs to be signed with a (certified) key from a conventional signature scheme. However, at some point the last set of keys is used, and new keys need to be generated and distributed (over a secure channel).

Clock synchronization between sender and receivers is required not only initially before the start of the communication but throughout the whole communication. This underlying assumption—synchronized sender and receiver clocks—may be violated by delay attacks, which we discuss in detail in Section 10.1. If the desynchronization between sender and receiver clocks is large enough, an adversary may know the disclosed key while the receiver accepts messages signed with that key as timely—effectively breaking the whole authentication scheme.

TESLA meets the performance requirements perfectly, since it has minimal communication overhead of roughly one MAC per packet. Furthermore, it tolerates packet loss (due to the use of a one-way chain), and resists collusion of receivers. Still, TESLA requires receivers to buffer messages (until the sender discloses the signing key and the receivers can verify the authenticity of the buffered messages). It is therefore vulnerable to DoS attacks on the receiver by flooding with bogus packets. Verifying the authenticity of a message consists not only in verifying the MAC but also checking whether the key belongs to the correct time interval. Otherwise, it would be trivially possible for an adversary to drop the original message and wait for the sender to disclose the key in order to forge messages on behalf of the sender.

Improvements to TESLA Perrig et al. suggested several modifications to their original TESLA proposal[54]. One modification is to allow receivers to authenticate most packets as soon as they arrive by replacing receiver buffering with

[51] Perrig et al., "Efficient authentication and signing of multicast streams over lossy channels".

[52] Adrian Perrig et al. "The TESLA broadcast authentication protocol". In: *RSA CryptoBytes* 5 (2002).

[53] Perrig et al., *Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction*.

[54] Adrian Perrig et al. "Efficient and Secure Source Authentication for Multicast". In: *Network and Distributed System Security Symposium, NDSS*. 2001, pp. 35–46.

sender buffering. In this way, the potential DoS attack against the receivers' buffers is prevented as receivers can immediately authenticate messages without the need to buffer them. Staggered TESLA[55] aims to decrease TESLA's susceptibility to DoS attacks, in which the receivers' buffers are filled until no more packets can be accepted. For this purpose, buffer utilization of receivers is improved so that bogus packets can be dropped faster by supporting intermediate security levels between "non-authenticated" and "fully-authenticated".

Another problem of the original TESLA proposal is that long time intervals cause long authentication delays. If the time interval is reduced such that it is smaller than the network delay, most messages will be dropped (for security reasons as the key could have been public before arrival). Perrig et al. propose to run multiple TESLA instances in parallel. These TESLA instances use different time intervals and one-way key chains, and receivers need to select which instance is most appropriate for them.

Despite all improvements, TESLA still requires the first packet to be signed by a conventional signature scheme in order to guarantee that the keys in the one-way chain have indeed been issued by the claimed sender. Perrig et al. designed μTESLA[56] for environments where sender and receivers are too resource constrained to use conventional signatures and therefore use a secure point-to-point link to bootstrap the key chain. Liu et al. propose an improvement to TESLA that neither requires a conventional signature nor point-to-point communication[57,58].

Bergadano Bergadano, Cavagnino, and Crispo published an authentication scheme called Chained Stream Authentication (CSA)[59] conceptionally similar to TESLA, also in 2000. They use time as source for asymmetry and a one-way key chain as well. However, CSA only sends a single message per time interval. The scheme's throughput is therefore significantly limited. Since time is used as source for asymmetry, the scheme also requires the sender and receiver clocks to be synchronized. Receivers need to buffer one packet until the verification key is distributed and the authenticity of the packet can be verified. The scheme is robust to packet loss. Since the length of the one-way chains is fixed, new chains need to be announced periodically (over a secure channel), which introduces additional communication overhead.

BiBa MTS The BiBa MTS scheme[60] builds upon the BiBa OTS scheme and uses time as source of asymmetry and therefore assumes sender and receiver clocks to be synchronized. Communication is divided into time intervals and one-way chains are employed that extends BiBa OTS to BiBa MTS such that receivers can verify the authenticity of messages and can recover if packets were lost. Since the lengths of the one-way chains needs to be fixed in advance, new keys need to generated and distributed periodically (over a secure channel). Furthermore, BiBa's communication overhead is large, and packets need to be buffered at the receiver before verification. Nevertheless, the scheme is resistant against collusion of receivers and provides immediate signing of packets. Unlike other MTS schemes, BiBa requires the maximum number of messages within a time period to be known in advance. Since the sender cannot disclose more than a certain number of balls in each interval, the sender has to use multiple BiBa instances to meet the application's packet send rate.

[55] Qing Li and W. Trappe. "Staggered TESLA: a multicast authentication scheme resistant to DoS attacks". In: *IEEE Global Telecommunications Conference (GLOBECOM)*. Nov. 2005. DOI: 10.1109/GLOCOM.2005.1577934.

[56] Adrian Perrig et al. "SPINS: Security Protocols for Sensor Networks". In: *Wireless Networks*. 2001, pp. 189–199.

[57] Donggang Liu and Peng Ning. "Multilevel μTESLA: Broadcast Authentication for Distributed Sensor Networks". In: *ACM Transactions Embedded Computing Systems* 3.4 (Nov. 2004), pp. 800–836. ISSN: 1539-9087. DOI: 10.1145/1027794.1027800.

[58] D. Liu et al. "Practical broadcast authentication in sensor networks". In: *International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*. July 2005, pp. 118–129. DOI: 10.1109/MOBIQUITOUS.2005.49.

[59] F. Bergadano, D. Cavagnino, and B. Crispo. "Individual single source authentication on the MBONE". in: *IEEE International Conference on Multimedia and Expo (ICME)*. 2000, pp. 541–544. DOI: 10.1109/ICME.2000.869659.

[60] Perrig, "The BiBa One-time Signature and Broadcast Authentication Protocol".

Computational efficiency is analyzed in detail by Law et al.[61]. Mitzenmacher and Perrig proposed Powerball[62] that aims to improve BiBa's verification efficiency.

TV-HORS In 2009, Wang et al. proposed a method called Time-Valid OTS (TV-OTS)[63] that can create an MTS scheme from any OTS scheme so that many packets can be signed securely (instead of just one). To this end, the authors propose to associate multiple one-time key pairs with a one-way chain. The authors furthermore propose to extend one-time keys to be used multiple times such that more packets can be signed with the same number of keys. However, exposed OTSs leak information about the secret key and therefore reduce the difficulty of forging a signature. Wang et al. argue that the signatures are only exposed for a limited time and the resulting MTS scheme can be considered secure when making careful assumptions about the attacker's resources.

Still, signature size and therefore communication overhead is large (because of the use of the OTS scheme). To reduce the signature size and computational cost, the authors suggest to only sign a part of a message's hash. Again, this reduces security as an adversary now needs to find only a partial hash collision. Since Time Valid HORS (TV-HORS) assumes that sender and receiver clocks are loosely synchronized, the time a signature is exposed to an adversary can be controlled. By estimating the adversary's computational power, it can be assured that no partial collision can be found with non-negligible probability (during the time span a key is valid).

To create TV-HORS, Wang et al. first use the TV-OTS model to extend the HORS OTS scheme to a MTS scheme such that the same key can be used to sign several messages. Instead of another OTS scheme, the authors use HORS because of its relatively small signature size. Fast message signing and verification (roughly 8k packets per second[64]) is achieved. It supports immediate authentication of messages, is resistant to collusion of clients, and tolerates packet loss. However, TV-HORS does assume the sender receiver clocks to be synchronized. The precision of clock synchronization between receivers and sender can be configured; if less precise clock synchronization is assumed, communication overhead increases. For this reason, TV-HORS comprises a trade-off between communication overhead and the required precision of clock synchronization. Nevertheless, compared to TESLA, TV-HORS is less sensitive to delay attacks. Since TV-HORS involves a k-time signature scheme, the frequent key updates require a secure out-of-band channel between sender and each receiver throughout the communication. Besides TV-HORS, other notable MTS schemes that base on HORS are Park and Cho's scheme[65], Tunable Signing and Verification (TSV)[66], TSV+[67], and HORSIC[68].

8.3.3 *Unrestricted-Time High-Speed Signing*

In addition to the six state-of-the-art classes of data origin authentication schemes for group communication, secret-information asymmetry, deferred signing, signature propagation, signature dispersal, OTS, and MTS, we suggest a new class: unrestricted-time high-speed signing. The unrestricted-

[61] Law et al., "Comparative Study of Multicast Authentication Schemes with Application to Wide-area Measurement System".

[62] Michael D Mitzenmacher and Adrian Perrig. "Bounds and improvements for BiBa signature schemes". In: (2002).

[63] Wang et al., "Time Valid One-Time Signature for Time-Critical Multicast Data Authentication".

[64] Wang et al., "Time Valid One-Time Signature for Time-Critical Multicast Data Authentication".

[65] Yongsu Park and Yookun Cho. "Efficient one-time signature schemes for stream authentication". In: *Journal of Information Science and Engineering* (2006).

[66] Qinghua Li and Guohong Cao. "Multicast Authentication in the Smart Grid With One-Time Signature". In: *IEEE Transactions on Smart Grid* 2.4 (Dec. 2011), pp. 686–696. ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2138172.

[67] Yee Wei Law et al. "WAKE: Key management scheme for wide-area measurement systems in smart grid". In: *Communications Magazine, IEEE* 51.1 (2013), pp. 34–41.

[68] Jaeheung Lee et al. "HORSIC: An efficient one-time signature scheme for wireless sensor networks". In: *Information Processing Letters* 112.20 (Oct. 31, 2012), pp. 783–787. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2012.07.007.

time high-speed signing class employs recently proposed high-speed signature schemes that offer previously unrivaled performance. For this reason, we argue to sign every packet independently in unrestricted-time high-speed signing despite the common assumption that it would be impractical due to the computationally cost. The particular performance of high-speed signature schemes in terms of computational efficiency and communication overhead is assessed in Chapter 9.

In unrestricted-time high-speed signing, there is no need to do trade-offs like in other classes of data origin authentication schemes because of the high computational efficiency and low communication overhead (as to be assessed in Chapter 9). Unrestricted-time high-speed signing provides immediate signing and immediate verification as neither the sender nor the receivers need to buffer packets. It provides collusion resistance since every receiver has identical information, the sender's public key. Authentication schemes cannot be entirely independent of time since the validity of the sender's public key needs to be ensured. However, while the other classes that follow the same approach of designing fast authentication schemes depend on a clock synchronization precision roughly in the order of seconds, the unrestricted-time high-speed signing class' dependency is in the order of months and therefore practically as independent as it can get. Furthermore, unrestricted-time high-speed signing provides resistance to packet loss as each packet carries independent authentication information. It is important that data origin authentication schemes provide authenticity of received packets if other packets were lost. Tolerance to packet loss is not for free, however, but it can be achieved when each packet carries its own authentication information (independent of other packets), which can only be justified as long as fast cryptographic mechanisms are used. The only property that is not satisfied by unrestricted-time high-speed signing is information-theoretical security, which can only be provided by schemes from the secret-information asymmetry class (that are not generally applicable).

Furthermore, we note that unrestricted-time high-speed signing is not the top performer for every single requirement (for example, TESLA is computationally more efficient). Our aim, however, is not to find the top performer for one particular requirement but a scheme that is generally applicable for secure group communication in critical infrastructures, and that is unrestricted-time high-speed signing.

8.4 Summary

Analyzing the timelines of publications of schemes in various classes (Figs. 8.1, 8.2, 8.3, 8.5, 8.7, and 8.8), it becomes apparent that specific classes are hardly researched anymore. Nevertheless, new applications such as sensor data collection (generally and in Smart Grids specifically) or streaming services, for example, strove to research MTS schemes. Seeking a generically suitable data origin authentication scheme, we evaluated various schemes from each class.

Each class of state-of-the-art data origin authentication schemes comprises a trade-off from a specific point of view, as summarized in Tables 8.1, 8.2, and 8.3. Secret-asymmetry schemes trade off information-theoretical security against collusion resistance, which means that they protect against adver-

saries with unbounded computational resources on the one hand, but at the same time are prone to receivers who collude in order to impersonate the sender. Deferred signing schemes trade off online computational resources against communication overhead and offline computational resources. Signature propagation schemes trade off computational efficiency and communication overhead against packet loss resistance as they rely on the successful reception of signature packets—from the moment a signature packet is missing the receiver cannot authenticate any more packets. Signature dispersal schemes trade off packet loss resistance against computational efficiency and immediate signing and verification such that the sender and the receivers need to wait before they can sign and verify packets, respectively, which is a drawback to applications with real-time requirements. OTS schemes can only sign a single message securely and continuously require a secure out-of-band channel for key distribution. MTS schemes trade off computational efficiency against a secure out-of-band channel for key distribution as well (and to a lesser extent also against independence of clock synchronization). In unrestricted-time high-speed signing, there is no need to do trade-offs like in other classes of data origin authentication schemes, which makes it suitable as general purpose schemes for securing group communication—not restricted to but especially in critical infrastructures.

Class	Scheme	Computational efficiency	Low communication overhead	Immediate signing	Immediate authentication
Secret-Information Asymmetry	k-MAC	+	+	+	+
	Desmedt et al.	-	-	+	+
Deferred Signing	Offline/online signing	-	--	+	+
	Offline/online k-time signing	~	~	+	+
Signature Propagation	Offline chaining	++	+	+	+
	Stream Signing	-	-	+	+
	EMSS	+	~	+	-
	RLH	~	~	+	-
	Augmented Chain Technique	+	~	-	-
Signature Dispersal	Wong & Lam	+	-	-	+
	SAIDA	-	-	-	-
	SAIDA improvements	~	~	-	-
	Tartary et al.	~	~	-	-
OTS	BiBa OTS	+	--	+	+
	HORS	++	--	+	+
MTS	BiBa MTS	+	~	+	-
	TESLA	++	+	+	-
	CSA	++	+	+	-
	TV-HORS	+	~	+	+
Unrestricted-Time High-Speed Signing	EdDSA / MQQ-SIG	+	+	+	+

Requirement is either: strongly satisfied (++), satisfied (+), somewhat satisfied (~), dissatisfied (-), or strongly dissatisfied (--).

Table 8.1: Theoretical performance evaluation of data origin authentication schemes.

Class	Scheme	Collusion resistance	Information-theoretical security
Secret-Information Asymmetry	k-MAC	−−	−
	Desmedt et al.	−−	+
Deferred Signing	Offline/online signing	+	−
	Offline/online k-time signing	+	−
Signature Propagation	Offline chaining	+	−
	Stream Signing	+	−
	EMSS	+	−
	RLH	+	−
	Augmented Chain Technique	+	−
Signature Dispersal	Wong-Lam	+	−
	SAIDA	+	−
	SAIDA improvements	+	−
	Tartary et al.	+	−
OTS	BiBa OTS	+	−
	HORS	+	−
MTS	BiBa MTS	+	−
	TESLA	+	−
	CSA	+	−
	TV-HORS	+	−
Unrestricted-Time High-Speed Signing	EdDSA / MQQ-SIG	+	−

Requirement is either: strongly satisfied (++), satisfied (+), somewhat satisfied (∼), dissatisfied (−), or strongly dissatisfied (−−).

Table 8.2: Theoretical security evaluation of data origin authentication schemes.

Class	Scheme	Resistance against packet loss	Independence of clock synchronization	Secure channel only initially required	Support for a wide range of applications
Secret-Information Asymmetry	k-MAC	+	+	-	-
	Desmedt et al.	+	+	-	-
Deferred Signing	Offline/online signing	+	+	-	+
	Offline/online k-time signing	-	+	-	+
Signature Propagation	Offline chaining	--	+	+	-
	Stream Signing	--	+	+	+
	EMSS	∼	+	+	+
	RLH	∼	+	+	+
	Augmented Chain Technique	∼	-	+	+
Signature Dispersal	Wong-Lam	+	+	+	+
	SAIDA	∼	+	+	+
	SAIDA improvments	∼	+	+	+
	Tartary et al.	∼	+	+	+
OTS	BiBa OTS	+	+	-	+
	HORS	+	+	-	+
MTS	BiBa MTS	+	--	-	-
	TESLA	+	--	-	+
	CSA	+	--	-	+
	TV-HORS	+	∼	-	+
Unrestricted-Time High-Speed Signing	EdDSA / MQQ-SIG	+	+	+	+

Requirement is either: strongly satisfied (++), satisfied (+), somewhat satisfied (∼), dissatisfied (-), or strongly dissatisfied (--).

Table 8.3: Theoretical robustness evaluation of data origin authentication schemes.

9

Assessment of Unrestricted-Time High-Speed Signing

This chapter has been published in part in:

Robert Annessi, Tanja Zseby, and Joachim Fabini. "A new Direction for Research on Data Origin Authentication in Group Communication". In: *International Conference on Cryptology and Network Security (CANS)*. Springer, 2017. DOI: 10.1007/978-3-030-02641-7_26. Referred to as "[I]". Adapted with permission from Springer Nature Customer Service Centre GmbH. © Springer Nature 2018.

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Robert Annessi, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252. Referred to as "[IV]". © ACM 2018.

In unrestricted-time high-speed signing, packets are signed independently with a digital signature scheme, despite the common assumption that it would be impractical due to the computationally cost. This assumption, however, only holds for conventional but not for novel high-performance signature schemes, which have been proposed in recent years and offer previously unrivaled performance. Employing novel high-performance signature schemes can mitigate the negative performance impact perceived from conventional schemes as we will assess in this chapter.

To this end, signature schemes are needed that offer previously unrivaled performance such as Ed25519[1], an elliptic-curve signature scheme "*carefully engineered at several levels of design and implementation to achieve very high speed without compromising security*"[2], or MQQ-SIG[3] a signature scheme based on MQ quasigroups. Both schemes are designed to provide fast signing and verification operations. Since many MQ signature schemes have been broken (including MQQ-SIG[4]) and some of them have been fixed and broken again, it is safe to say that MQ schemes comprise serious security challenges. For

[1] Bernstein et al., "High-speed high-security signatures".

[2] Bernstein et al., "High-speed high-security signatures".

[3] Danilo Gligoroski et al. "MQQ-SIG". in: *Trusted Systems*. Springer, 2011, pp. 184–203.

[4] Jean-Charles Faugere et al. "A polynomial-time key-recovery attack on MQQ cryptosystems". In: *IACR International Workshop on Public Key Cryptography*. Springer, 2015, pp. 150–174.

this reason, we need to stress that the use of MQQ-SIG specifically is not recommended in practice. Nevertheless, we include MQQ-SIG in our evaluation since MQ schemes have attractive properties (specifically post-quantum security), and MQQ-SIG is one of the fastest MQ signature schemes. Furthermore, we hope that highlighting group communication use-cases spurs future research on MQ signature schemes.

Performance Evaluation

To evaluate whether our new class of data origin authentication schemes can indeed deliver the required computational efficiency and low communication overhead, we test the two high-performance digital signature schemes mentioned before (both at a conjectured security level of 128 bit): Ed25519 and MQQ-SIG. In an experiment, we measured the computational efficiency of these high-performance signature schemes on Commercial Off-The-Shelf (COTS) hardware, an Intel Celeron CPU clocked at 2.26 GHz running Debian Linux 8 32-bit. We disabled Intel's Hyper-threading and Turbo Boost, CPU-frequency scaling, and CPU-sleep states to not interfere with the measurement. Ed25519 signed and verified about 13k packets per second and has a communication overhead of 64 B per packet. MQQ-SIG signed and verified over 36k packets per second with a communication overhead of 32 B per packet. In this way, unrestricted-time high-speed signing outperforms the best state-of-the-art data origin authentication scheme, TV-HORS from the MTS class, which can sign and verify only 5k packets per second with a communication overhead of 106 B per packet[5]. Admittedly, the measurements for TV-HORS and the signature schemes were not conducted under the exact same conditions. The goal, however, was to check whether unrestricted-time high-speed signing can provide the desired computational efficiency and communication overhead—and this turns out to be very much the case. Table 9.1 summarizes the measurement results.

[5] Teklemariam Tesfay and Jean-Yves Le Boudec. "Experimental Comparison of Multicast Authentication for Wide Area Monitoring Systems". In: *IEEE Transactions on Smart Grid* (2017). ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2017.2656067.

Scheme	Signing and verification	Overhead
Ed25519	13k packets / s	64 B / packet
MQQ-SIG	36k packets / s	32 B / packet
TV-HORS	5k packets / s	106 B / packet

Table 9.1: Computational efficiency and communication overhead of high-speed signature schemes compared to TV-HORS. Adapted from [I] with permission from Springer Nature Customer Service Centre GmbH. © Springer Nature 2018.

As highlighted in Chapter 8, each class of data origin authentication schemes comprises a trade-off from a specific point of view. In unrestricted-time high-speed signing, there is no need to do trade-offs like in other classes of data origin authentication schemes because of the high computational efficiency and low communication overhead. For this reason, unrestricted-time high-speed signing is suitable as general purpose schemes for securing group communication (in critical infrastructures).

10
Secure Clock Synchronization

This chapter has been published in part in:

Robert Annessi, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418. Referred to as "[II]". © IEEE 2017.

Robert Annessi, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR]. Referred to as "[III]".

Robert Annessi, Joachim Fabini, Felix Iglesias, and Tanja Zseby. *Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization*. 2018. arXiv: 1811.08569 [cs.CR]. Referred to as "[VIII]".

As highlighted in Chapter 2, clock synchronization protocols have become an essential building block of critical infrastructure applications that rely on a precise notion of time. Group communication is important for clock synchronization protocols, but how to secure multicast clock synchronization remains an open research question. As will be highlighted in this chapter, it is essential for data origin authentication schemes to be independent of precisely synchronized clocks in order to secure multicast clock synchronization. Furthermore, immediate signing is essential—otherwise, clock synchronization precision is negatively impacted. Surprisingly, computational efficiency is only import for securing the one-step mode in clock synchronization as will be highlighted in Section 10.2.

The NTS drafts[1,2,3] propose to employ TESLA[4,5,6] to address the data origin authentication problem. Since TESLA was the favored approach by the IETF NTP group and is highly rated by the P1588 Security Subcommittee for securing the next version of PTP, we evaluate it in greater detail. While NTS prevents substitution attacks and pre-play attacks, delay attacks were not addressed specifically. Replay protection is realized with TESLA but this requires that only one message is sent per time interval which is rather inefficient and may facilitate a simple DoS attack on the receivers' buffers. TESLA meets the performance requirements perfectly, since it requires few computational resources and has minimal communication overhead of just one MAC per packet. It furthermore tolerates packet loss and resists collusion of receivers. Still, we find TESLA unsuitable to secure multicast clock synchronization messages because it is vulnerable to message delay attacks in the context of clock synchronization.

[1] Sibold, Roettger, and Teichel, *Network Time Security*.

[2] Sibold, Roettger, and Teichel, *Using the Network Time Security Specification to Secure the Network Time Protocol*.

[3] Sibold et al., *Protecting Network Time Security Messages with the Cryptographic Message Syntax (CMS)*.

[4] Perrig et al., "Efficient authentication and signing of multicast streams over lossy channels".

[5] Perrig et al., "The TESLA broadcast authentication protocol".

[6] Perrig et al., *Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction*.

In TESLA, time is partitioned into predefined intervals. During each interval, the master sends signed messages that the slaves buffer. At the end of each interval, the master discloses the signing key such that the slaves can verify the authenticity of the buffered messages. Verifying the authenticity of a message consists not only of verifying the MAC but also of checking whether the key belongs to the correct time interval. Otherwise, it would be trivially possible for an adversary to drop the original message and wait for the master to disclose the key in order to forge messages on behalf of the master.

As outlined in Chapter 6, data origin authentication schemes cannot provide protection against every attack type: message delay attacks can even be conducted when an authentication scheme is in place. Assuming synchronized master and slave clocks is dangerous for securing clock synchronization messages because it includes circular reasoning and creates a dependency on a common notion of time[7]. In case the dependency on a common notion of time is violated, the security of the authentication scheme breaks.

[7] Malhotra and Goldberg, "Attacking NTP's Authenticated Broadcast Mode".

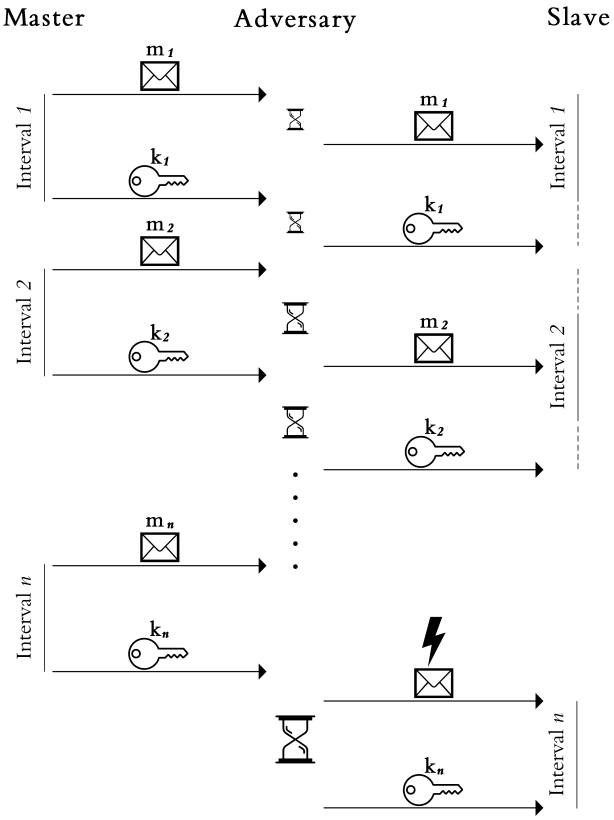

Figure 10.1: Delay attack on TESLA as part of a clock synchronization protocol.
Adapted from [II]. © IEEE 2017.

A message delay attack on TESLA in the context of clock synchronization is depicted in Figure 10.1. In the first interval, the adversary delays the clock synchronization and key disclosure messages by a short time such that the slave sets its clock backwards. For this reason, the slave's perception of the interval is slightly shifted. This difference in the slave's perception increases as the adversary continues to delay messages maliciously (Interval 2). The small-scale delays accumulate into a large-scale clock shift eventually so that the adversary gets access to the key before the end of the interval (Interval n). For this reason, the adversary can then generate authentication information for arbitrary messages that the slave will accept as valid.

To mitigate delay attacks on TESLA, NTS proposes two distinct strategies: (1) introduce unicast messages to the TESLA protocol such that TESLA's susceptibility to delay attacks can be mitigated[8], and, (2) leaving the slave vulnerable[9]. We argue that both are failure criteria: (1) the goal of the additional check ("keycheck") is to guarantee "*to the client that the key belonging to the respective TESLA interval communicated in the exchange had not been disclosed before*"[10]. To verify that the multicast message has not been delayed by some malicious entity, the slave is expected to establish a unicast connection to the master after each multicast message. But establishing a unicast connection after the reception of each multicast packet defeats the purpose of using group communication in the first place. (2) And leaving the slave vulnerable to delay attacks by proposing this additional keycheck to be optional effectively negates the whole authentication procedure. We therefore argue that both countermeasures are unsuitable.

[8] Kristof Teichel, Dieter Sibold, and Stefan Milius. "An attack possibility on time synchronization protocols secured with TESLA-like mechanisms". In: *Information Systems Security*. Springer, 2016, pp. 3–22.

[9] Sibold, Roettger, and Teichel, *Using the Network Time Security Specification to Secure the Network Time Protocol*.

[10] Sibold, Roettger, and Teichel, *Network Time Security*.

Notation Throughout this chapter, we will use the notation from Table 10.1. The specific values for d_{min} and d_{max} depend on the network (local private network or public Internet) and on the quality of service measures in place, as to be explained in Section 10.1[11].

[11] If the adversary is assumed to be able to transfer packets with negligible delay (as done in Chapter 4), d_{min} should be set to 0.

Symbol	Description
d_{min}^{MS}	The minimum one-way delay from master to slave.
d_{min}^{SM}	The minimum one-way delay from slave to master.
RTD_{max}	The maximum round-trip delay that is accepted in a clock synchronization interval.
ρ	The maximum drift of the slave clock relative to that of the master.
t_{arr}	The time of a successful clock synchronization interval.
t_{last}	The time when the slave corrected its clock last (initially set to ∞).

Table 10.1: Notation used for delay attacks and SecureTime.

10.1 Delay Attacks

Whenever clock synchronization messages are neither encrypted nor integrity-protected, an adversary can attack clock synchronization protocols in the value domain, i.e., the adversary modifies the timestamp values included in the messages. This is a well-studied field and various countermeasures have been proposed to secure clock synchronization against attacks in the value domain such as encrypting the communication (with MACsec, IPsec, or TLS) or employing digital signatures to ensure the integrity and authenticity of the communication.

To conduct an attack in the time domain an adversary intercepts clock synchronization messages and delays them artificially for some time before forwarding[12]. The maliciously introduced delay can be constant, variable, or even random, and the slave clock can be manipulated this way[13]. Since the clock synchronization protocol has no information on the underlying communication network, one fundamental prerequisite and assumption of PTP is symmetric delay between master and slave. I.e., the OWD from master to slave is identical to the OWD from slave to master. Delay attacks exploit this

[12] Theoretically delay attacks can also be conducted by accelerating messages (instead of delaying them). To conduct such acceleration attack, the adversary needs to delay all messages by default and selectively forward some messages with less delay in order to achieve an acceleration effect. Another option to accelerate messages is to route them through faster paths if such option is available to the attacker.

[13] T. Mizrahi. *Security Requirements of Time Protocols in Packet Switched Networks*. RFC 7384 (Informational). RFC. Fremont, CA, USA: RFC Editor, Oct. 2014. DOI: 10.17487/RFC7384. URL: https://www.rfc-editor.org/rfc/rfc7384.txt.

assumption of symmetric delays by maliciously introducing asymmetry such that the slave synchronizes to a false time.

In order to conduct a delay attack, the attacker maliciously manipulates one of the two messages that are crucial to clock offset measurement and delay measurement: SYNC and DELAY_REQUEST. While other network-based attacks are important as well, delay attacks and especially countermeasures against delay attacks on clock synchronization have not been studied in required depth yet. This section focuses on this gap in research—delay attacks and their impact on clock synchronization's precision.

As we will show in Subsections 10.1.1 and 10.1.3, delay attacks are feasible despite security measures in place (i.e., traffic being encrypted and integrity-protected). Encrypting traffic is not sufficient, mainly because successful verification of a message's integrity only certifies the correctness of the sending time reported in the message but not its effective propagation time through the network[14]. For this reason, delay attacks can also be conducted on encrypted and integrity-protected traffic. The assumption of symmetric OWDs is essential to delay attacks as those exploit non-deterministic delays in communication networks.

[14] Katz, *Digital Signatures*.

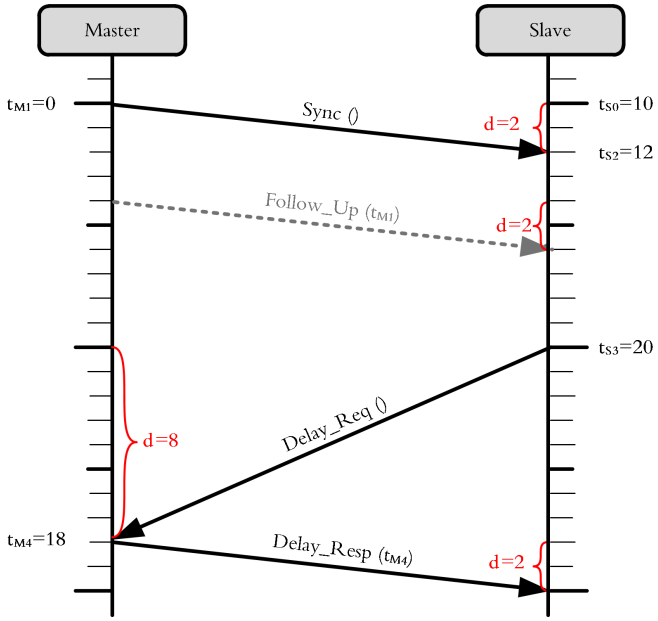

Figure 10.2: PTP clock synchronization with asymmetric delay. Adapted from [VIII].

Asymmetrically delayed SYNC and DELAY_REQUEST messages lead to miscalculated clock offset. The effect of asymmetrically delayed DELAY_REQUEST and SYNC messages on the calculated clock offset is depicted in Fig. 10.2 and 10.3, respectively. In both figures, the OWD from master to slave and from slave to master is supposed to be symmetric and should equal 2 time units. The slave therefore calculates the clock offset relative to the master precisely as 10 (according to Eq. 2.7). In Fig. 10.2, however, the DELAY_REQUEST message from slave to master is delayed such that it takes 8 time units (instead of 2). For this reason, the slave miscalculates the clock offset as 7 (according to Eq. 2.7) so that the slave clock remains 3 time units ahead of the master's. If, on the other hand, the SYNC message takes 8 time units instead of 2 (Fig. 10.3), the slave miscalculates the clock offset (again according to Eq. 2.7) as 13 so that the slave corrects its clock too much and its clock is then behind the master's.

Such asymmetric delays are an inherent part of packet-switched communication networks as transmission delays, propagation delays, queuing delays, and processing delays are practically never entirely symmetric[15]. Sometimes, asymmetry is even intentional like with Ethernet where cables are asymmetric by design to reduce far end crosstalk[16]. Such static asymmetry can be compensated, but asymmetry may also be maliciously introduced such that it is non-predictable and therefore cannot be compensated straightforwardly as to be shown in Subsections 10.1.1 and 10.1.3.

[15] In order to have symmetric delays the bit rate of all links in the path must be identical, routing in both directions must be identical, traffic flow in both directions must be identical, and both end systems must behave identically and must be in the same state.

[16] Yang, An, and Yu, "On time desynchronization attack against IEEE 1588 protocol in power grid systems".

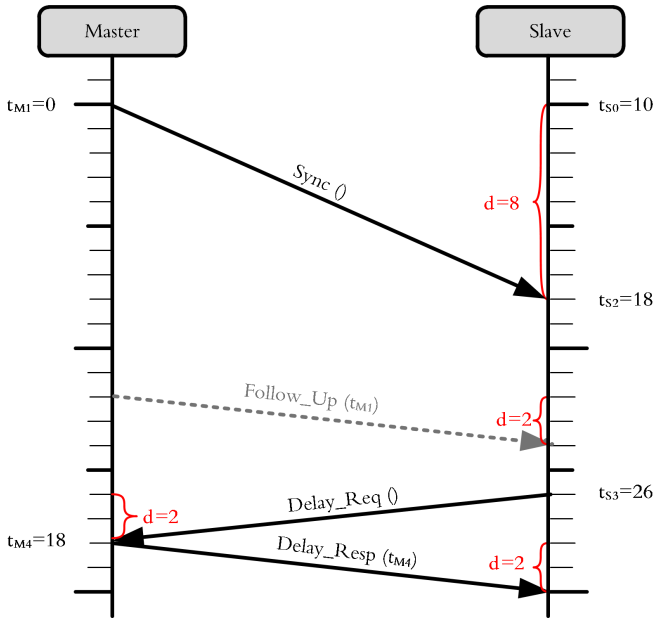

Figure 10.3: PTP clock synchronization where SYNC message is affected by additional delay.

Adapted from [VIII].

10.1.1 Selective Message Delay Attacks

To secure communications over untrusted networks the entire communication is commonly encrypted and authenticated, for instance with IPsec. For this reason, we tested delay attacks with IPsec in tunnel mode. We have tested selective message delay attacks successfully against commercially available systems like routers or protections switches that tunnel PTP using security protocols different from IPsec, too. Methods and conclusions are identical to the ones presented for IPsec tunnels for all tested systems. In such a scenario, the attacker has access only to encrypted traffic, which means that there is no information available about protocols, source and destination ports, nor the IP addresses of the real endpoints (but only those of the tunnel routers).

In this subsection, we show that clock synchronization messages can be reliably identified in an encrypted traffic stream with reasonable effort. This identification of (encrypted) clock synchronization messages builds the foundation for selective message delay attacks. For this purpose, we aim to answer the following questions:

- Are there any (statistical) properties of PTP traffic that can be used to identify PTP messages within encrypted traffic?
- Can PTP traffic be modified such that delay attacks can be prevented or at least be mitigated?

- Can encryption schemes provide reasonable security against selective message delay attacks?

To answer whether it is possible to identify both PTP traffic in general and specific types of PTP messages in encrypted traffic, we conduct a statistical traffic analysis of PTP traffic. In this analysis, we identify several properties of PTP traffic that build the foundation for selective message delay attacks. We furthermore implement selective message delay attacks on actual devices to show the feasibility of our proof of concept in practice.

PTP Traffic Analysis

In order to make a slave adhere to a false time, either the SYNC or the DELAY_REQUEST message need to be delayed by the adversary (as pointed in the beginning of Section 10.1). Depending on the delaying of either SYNC or DELAY_REQUEST, the slave's notion of time is going to be behind or ahead of the master's, respectively. When traffic is encrypted, traffic analysis is limited to a restricted set of features: time, packet length, source, and destination. It is therefore not obvious which message type is observed in the network. Only the features can potentially be used to find out whether a packet is a PTP packet and which type of PTP packet it is. The statistical properties identified in PTP in both phases, i.e., clock offset measurement and delay measurement, are closely related to timing, packet length, and packet direction.

One PTP clock synchronization cycle (in two-step mode) consists of a series of four messages (as highlighted in Section 2.8): (1) A SYNC message from the master to the slave. (2) Another message (FOLLOW_UP) from the master to the slave. (3) A DELAY_REQUEST message in the reverse direction from the slave to the master, and (4) a message from the master to the slave (DELAY_RESPONSE). This series repeats at a fixed interval. Every two seconds we observe another PTP message (ANNOUNCE) from the master to the slave. This ANNOUNCE message is used for the Best-Master-Clock algorithm, which we do not focus on in this thesis.

Our traffic analysis revealed some specific properties of PTP traffic in two-step mode (and we expect similar results for PTP's one-step mode). These properties are the length of packets, the timing, and the direction of messages. The lengths of the packets are appealing as they are highly deterministic and mostly constant. The reason for this is that the designers of PTP wanted to avoid variation in transmission delays due to different packet lengths. Fig. 10.4 sketches the results of our PTP traffic analysis. Firstly, all messages from the master to a slave are of same length and the DELAY_REQUEST message in the reverse direction is either of equal length or slightly larger, which means that the length of a PTP message can be related to its direction. The specific lengths of the messages depend on the underlying communication protocols and on which layer the messages are observed. In our setup, the lengths of (unencrypted) PTP messages were 86 B and 96 B for SYNC, FOLLOW_UP, and DELAY_RESPONSE and DELAY_REQUEST, respectively (and 106 B for ANNOUNCE messages). In encrypted traffic, additional information is added to the packet by the encryption scheme, increasing the packet's length. The packet lengths observed were 138 B and 154 B in a test with IPsec encryption. Other encryption methods result in different lengths but the observed pattern

persists. Messages of length 154 B occur every 2 s, which corresponds to Announce messages. The remaining packets with a length of 138 B occur every 250 ms and in sets of 4 (which corresponds to the SYNC, FOLLOW_UP, DELAY_REQUEST, and DELAY_RESPONSE messages). The length of encrypted PTP messages are deterministic, as well, and therefore identifiable.

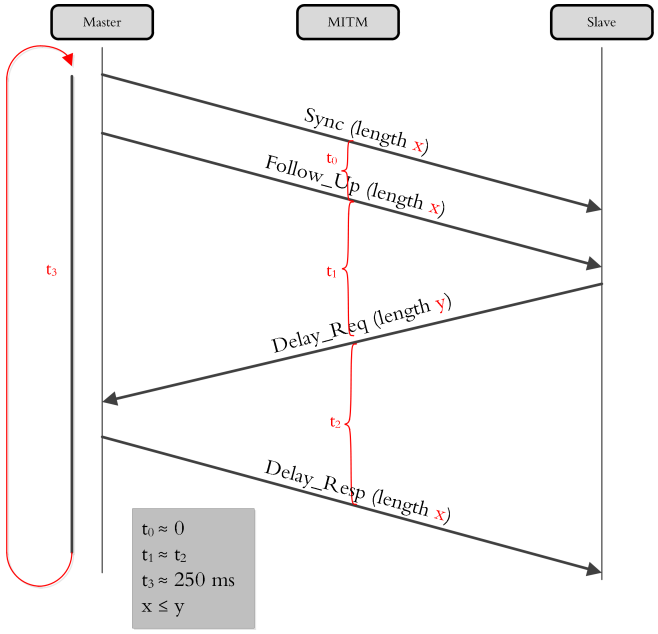

Figure 10.4: PTP traffic patterns in timing, length, and direction.
Adapted from [VIII].

Secondly, we observe that PTP messages follow a specific timing pattern. The FOLLOW_UP message is sent from the master with minimal delay (t_0) after the SYNC message, the time difference observed (t_1) between DELAY_REQUEST and FOLLOW_UP messages, and the time difference observed (t_2) between DELAY_REQUEST and DELAY_RESPONSE messages. From the adversary's point of view t_1 and t_2 are roughly identical, which is about the OWD between master and slave[17]. Because of their periodicity, the observed timings are also visible in encrypted traffic, as we will see later in this subsection.

Thirdly, the direction of the messages is fixed as long as the master and slave roles persist. And the observed pattern repeats periodically at a fixed interval (t_3) for as long as the clock synchronization service is running. The clock synchronization interval can be configured and was left to the default setting of 250 ms during our tests. Table 10.2 summarizes the results of our traffic analysis. Next, we show how these properties can be used to identify PTP traffic in a stream of encrypted traffic.

[17] t_1 and t_2 are roughly identical if the attacker is positioned equidistant in terms of time between master and slave. If the attacker is closer to either master or slave, the difference between t_1 and t_2 increases—visible by shifting the observation point in Fig. 10.4 to the left or to the right, respectively.

Identification of PTP messages in Encrypted Traffic

In order to conduct a selective message delay attack on an encrypted traffic stream, first the specific PTP messages need to be identified within the encrypted traffic without prior knowledge of the communications within that stream. To this end, we setup a proof of concept to verify that PTP message types can be identified within encrypted traffic. In a real-world scenario, the specific setup will always be different, and attackers may not have the plaintext communication available to figure out the parameters of the setup (i.e., the specific packet sizes, the PTP session interval, etc). Also, packet lengths

Attribution to paper coauthors

The experiments on identifying PTP messages in encrypted traffic with statistical methods (described in the joint paper [VIII] and Subsection 10.1.1) was conducted primarily by Félix Iglesias Vázquez.

Message type	Direction (master-slave)	Length (unencrypted/encrypted)	Observed time differences
SYNC	→	86 B / 138 B	t_3
FOLLOW_UP	→	86 B / 138 B	t_0
DELAY_REQUEST	←	96 B / 138 B	$t_1 \approx t_2 \approx$ slave → master one-way delay
DELAY_RESPONSE	→	86 B / 138 B	$t_2 \approx t_1 \approx$ master → slave one-way delay
ANNOUNCE	→	106 B / 154 B	fixed interval (ignored)

Table 10.2: Identified properties of PTP traffic. t_3 with regard to the last DELAY_RESPONSE message, i.e., the clock synchronization interval.
Adapted from [VIII].

differ in encrypted traffic (as there is additional data added by the encryption layer) but the basic properties (frequency, direction, timing, and the relation of the lengths) persist.

Using the results of our PTP traffic analysis, we wanted to know whether identifying PTP messages in encrypted traffic is feasible. For this purpose, we conducted some experiments, in which PTP communications were simulated. The simulator chooses random timings t_0, t_1, t_2, t_3, and random lengths x and y. The simulation takes two parameters: the noise level and the time to observe the traffic. Non-PTP packets are added to the simulated PTP communication, which reflects the probability to observe a non-PTP packet of random length every sampling interval (i.e., noise is added from a statistical perspective). Eventually, the proof of concept tries to estimate t_0, t_1, t_2, t_3, and the lengths x and y just by observing the communication.

This proof of concept relies on four assumptions:

1. Time is discretized with a sampling time of 1 ms.
2. There is only one packet per time bin.
3. The PTP communication pattern repeats over the entire observation period.
4. t_0, t_1, t_2, and t_3 are constant from the perspective of the sampling time.

Such assumptions draw some limitations for simulating communications, but they are reasonable since experiments are intended to be proofs of concept and not fully-fledged implementations. Real-world communications comprise additional complexities that might defy the proof of concept detector, but such situation can usually be faced by refining the detector (for example, by using recurrence analysis, Granger causality, or Markov models). Our goal is to show that the detection of the pursued time parameters and lengths is theoretically possible and feasible by applying methods based on statistics.

Our simulations show that packet lengths, directions, and timing are sufficient to separate PTP from other traffic and even to identify the particular type of PTP message so that the selective message delay attack can be conducted. If the noise level is increased, the observation time needs to be increased as well (as expected). Under our simulation conditions, with 99.9% non-PTP packets, we need to observe the communications for roughly 1000 seconds to reliably determine the particular times and lengths.

A challenging scenario for the detection would be the occurrence of periodic signals with similar properties (timing, lengths, and direction) within the

same traffic stream. We argue that the properties of PTP are very particular and the chances to encounter communications with similar properties in the same traffic are very low. In the light of our experimental results, we conclude that PTP traffic can be identified with high probability within encrypted network traffic. For this reason, selective message delay attacks can be conducted on PTP even when it is (supposedly) secured with state-of-the-art network encryption schemes such as IPsec.

Experimental Results

Based on the statistical properties we identified in the traffic analysis and the theoretical confirmation that these properties can be used to identify PTP messages within encrypted traffic, we implemented a PTP traffic detection and PTP message type identification on a real clock synchronization system[18]. We furthermore implemented selective message delay attacks on actual devices to show the feasibility of our proof of concept in a practical setting. Fig. 10.5 depicts the experimental setup we used to evaluate the feasibility of our proof of concept and to examine the effect of delay attacks on real PTP systems over untrusted networks. We used three Linux systems: one that runs as PTP master, another as PTP slave, and the third acts as network bridge. Master and slave were connected through an IPsec tunnel such that the bridge could only observe encrypted traffic. PTP master and slave both run PTPd version 2.3.1. Master and slave receive a GPS PPS signal but only the master clock is synchronized to it. With this setup, we can synchronize the master clock $\pm 10\,\mu s$ to UTC, which is not overly precise but enough to highlight the effect of selective message delay attacks. The slave clock is synchronized to the master via PTP. To compare the slave clock to the GPS PPS signal, the "ppstest" tool[19] was used.

[18] But without adding non-PTP traffic.

[19] https://github.com/redlab-i/pps-tools

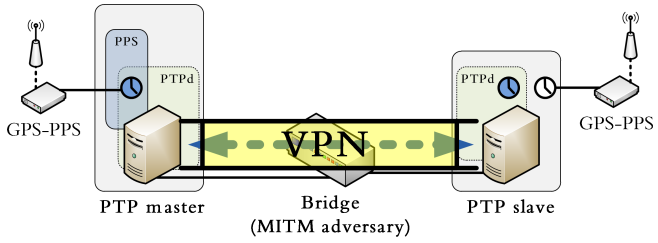

Figure 10.5: Experimental setup used to examine the effect of delay attacks on PTP over an untrusted network.

Adapted from [VIII].

On the network bridge, we implemented a MITM application that can delay specific packets using the identified properties. The bridge is implemented with "libnetfilter_queue"[20] so that packets are not only available in kernel space but also in user space, which facilitates easier classification and attack implementation. Since traffic is encrypted, a delay attack in the value domain is not possible because the timestamps within the packets cannot be modified. Instead, the attack only works in the time domain. In general, the adversarial application on the bridge aims to identify PTP messages and delay specific messages as soon as the selective message delay attack is started. We expect that the slave clock is then desynchronized from the master clock (after a short time, which his due to an averaging algorithm employed in PTPd).

[20] https://netfilter.org/projects/libnetfilter_queue/

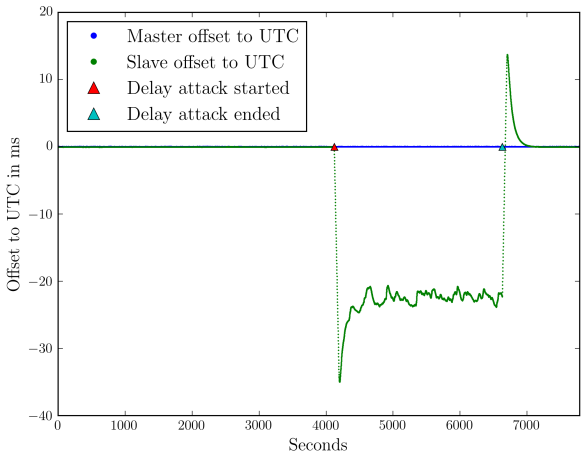

Figure 10.6: Offset to UTC during an selective SYNC message delay attack.

Adapted from [VIII].

Experiment 1: Delayed SYNC *messages* In the first experiment, we programmed our malicious application to delay all SYNC (and FOLLOW_UP) messages by 50 ms. We chose such large delay to stress that during such an attack the clock synchronization cannot be considered high-precision anymore. Fig. 10.6 shows the offset of the master and of the slave clock to UTC during normal operation and during the attack. The master clock is quite stable throughout the run of the experiment and not affected at all by the selective SYNC-message delay attack, as expected. The slave clock is affected, however, since the clock synchronization messages of the master are delayed maliciously. The slave clock spikes[21] shortly after the attack is started (at time 4122) and ended (at time 6632) and settles around −25 ms to UTC after a couple of seconds throughout the attack. The delay attack therefore operates as intended since the slave clock is around 25 ms behind the master's during the attack.

[21] Presumably, the overshooting in those spikes at the beginning and at the end of the attack is caused by the specific control algorithm implementation in PTPd.

Experiment 2: Delayed DELAY_REQUEST *messages* The same setup was used in a second experiment. This time, DELAY_REQUEST messages were delayed 50 ms by our adversarial application on the bridge (instead of SYNC and FOLLOW_UP messages). For this reason the slave clock during the attack is (roughly 25 ms) ahead of the master's clock as shown in Fig. 10.7.

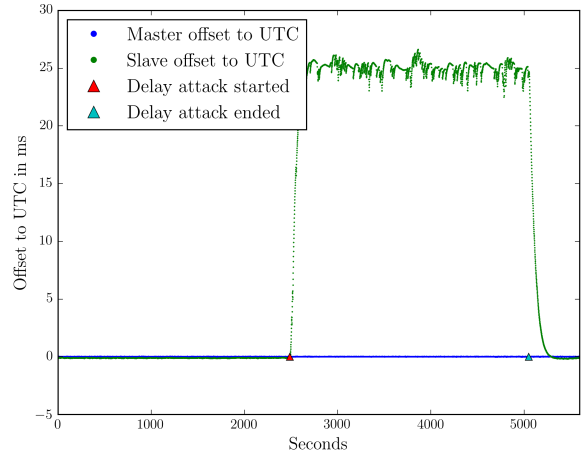

Figure 10.7: Offset to UTC during a selective DELAY_REQUEST message delay attack.

Adapted from [VIII].

Experiment 3: Incremental delay attack One may argue that the spikes in clock offset at the start and end of the attacks could raise a suspicion in security-critical environments. For this reason, we also implemented selective message delay attacks that incrementally add malicious delay. In that case (shown in Fig. 10.8), the attacker does not apply the full malicious delay from the moment the attack is started but instead increases the malicious delay with each clock synchronization interval incrementally. In this way, there is no more spike in the slave's clock offset when the attack starts. At time 3119 the incremental delay attack was started as DELAY_REQUEST messages were (increasingly) delayed by 1 ppm, i.e., a delay of 1 µs per second.

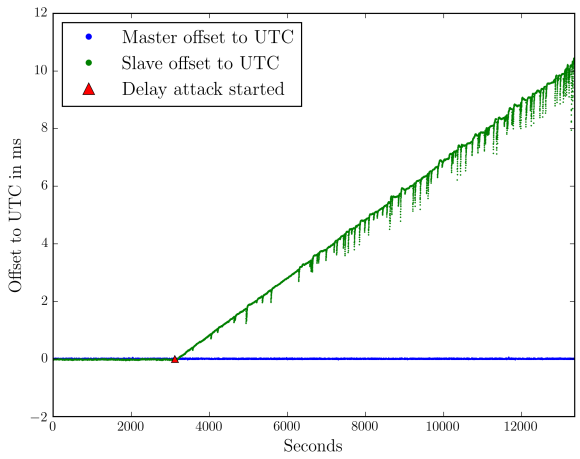

Figure 10.8: Offset to UTC during a incremental selective DELAY_REQUEST message delay attack.
Adapted from [VIII].

This incremental delay was deliberately chosen small (with 1 ppm) to highlight that such small delay is indistinguishable from delay variation while still having a significant effect on the clock offset eventually. Despite the small incremental delay, the slave clock is off more than 7 ms to UTC after two hours, which is completely unacceptable for time-critical systems that rely on a precise notion of time. However, the slave cannot notice the drift of its clock relative to the master's because of the incremental delay attack and is therefore convinced to be perfectly synchronized.

Discussion While the first two attacks may be detectable due to the spikes in the clock offset at the start and end of the attacks, the incremental attack cannot be detected easily. Moreover, we argue later in Subsection 10.1.4 that such incremental delay attack cannot be prevented at all (only be mitigated to some extent under specific circumstances). Although the attacker can neither observe the packets' contents, nor the real endpoints, nor ports, selective message delay attacks can be conducted successfully. This indicates that encryption alone cannot prevent selective message delay attacks on clock synchronization.

10.1.2 Countermeasures Against Selective Message Delay Attacks

In order to secure PTP against selective message delay attacks, encryption of the communication is not sufficient. But there exist two options to counter selective message delay attacks: (1) prevent traffic analysis, and (2) mitigate the actual attack. Both options will be discussed in this subsection.

Traffic Analysis Mitigation

Direction, timing, and packet length are sufficient to reliably identify PTP traffic and the specific types of PTP messages. In traffic analysis mitigation, the goal is to disturb traffic analysis. For this purpose, we need to make sure that the observable features entail no information that can be used to conduct an attack. These observable features are packet length, time, and direction.

While packet lengths are usually highly deterministic and constant, they can be hidden by padding to a fixed length. Such padding can even be implemented without changing the clock synchronization protocol. Encapsulating Security Payload (ESP) mode in IPsec, for example, supports payload padding up to 255 padding bytes[22], and the Extension Header could be used therefor in IPv6[23]. Alternatively or additionally Traffic Flow Confidentiality (TFC)[24] could be employed to ensure that all PTP messages have the same length. Alternatively, random lengths could be used, but this would increase variation of transmission delays (in addition to increasing bandwidth requirements), which is detrimental to the goal of achieving high-precision clock synchronization.

Nevertheless, padding alone may not be sufficient for traffic analysis mitigation, since information about the time and the direction could be enough to identify PTP message types (and therefore to conduct selective message delay attacks). The next feature that needs to be changed is the timing. Because of the periodicity of PTP messages discovered in our PTP traffic analysis, the specific messages can be identified reliably even in encrypted traffic. Changing the timing, however, can only be done from within the clock synchronization protocol and not by external mechanisms. The offset measurement and delay measurement could be separated, the offset correction not executed periodically but in random intervals, and there could be a random interval as well between SYNC and FOLLOW_UP messages. In this way, the timing properties of PTP could hardly be used anymore to identify PTP packets reliably under the assumption that sufficient cover traffic exists with similar timing properties. The major downside of this method is that it depends on the continuous existence of suitable cover traffic over the entire path from master to slave. This prerequisite of suitable cover traffic shifts the discussion to the well-researched area of traffic obfuscation in order to protect PTP from selective message delay attacks (even when traffic is encrypted). Traffic obfuscation is known to be very complex and its security highly depends on the specific threat model[25].

The last feature that is used to prepare selective message delay attacks is the direction of messages. As we have seen from the traffic analysis, the packets' directions are highly deterministic in PTP. However, there is no straightforward way to remove this feature, since network addresses are essential to communication networks.

Delay Attack Mitigation

In this subsection, we highlight a technique to mitigate delay attack that builds upon the replay protection of the encryption scheme. It needs to be stressed, however, that the technique is a mitigation only and cannot prevent the attacks. Table 10.3 summarizes the results.

As the name suggests, the replay protection of a network security protocol is supposed to protect against replay attacks (and has not been specifically

[22] S. Kent. *IP Encapsulating Security Payload (ESP)*. RFC 4303 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Dec. 2005. DOI: 10.17487/RFC4303. URL: https://www.rfc-editor.org/rfc/rfc4303.txt.

[23] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460 (Draft Standard). RFC. Obsoleted by RFC 8200, updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946, 7045, 7112. Fremont, CA, USA: RFC Editor, Dec. 1998. DOI: 10.17487/RFC2460. URL: https://www.rfc-editor.org/rfc/rfc2460.txt.

[24] P. Carlén. "Traffic Flow Confidentiality mechanisms and their impact on traffic". In: *2013 Military Communications and Information Systems Conference*. Oct. 2013, pp. 1–6.

[25] A special case here is that other periodic signals (that show similar properties to those identified in PTP) could be used to mitigate traffic analysis.

Countermeasure	Benefit	Drawbacks	Prevents attack
Random send and reply times	Confuses traffic analysis based on timings.	Requires protocol changes and depends on cover traffic.	No
Equal (or variable) message lengths	Confuses traffic analysis based on lengths.	Variable lengths reduce precision and depends on cover traffic.	No
Strict replay protection	Limits max. impact of the attack.	Not feasible in scenarios, in which reducing packet loss rate is more important than stricter replay protection.	No
Limiting OWDs	Limits max. impact of the attack.	Requires knowledge of the underlying communication network.	No

Table 10.3: Countermeasures against selective message delay attacks.
Adapted from [VIII].

designed to prevent delay attacks). At the same time, however, the replay protection also limits the maximum impact of selective message delay attacks since packets cannot be delayed arbitrarily. The impact of selective message delay attacks may be limited by maxing the encryption scheme's replay protection and assuring that a sufficient number of packets per clock synchronization interval are sent through the network as cover traffic. Replay protection usually works as follows: a sequence number is added to the packet, and the receiver accepts packets with strictly increasing sequence numbers only or with sequence numbers from a certain window to allow packets to overtake other packets in the network. While not specifically designed for this purpose, such replay protection also limits the impact of delay attacks since packets will not be accepted by the receiver if too many other packets have been received meanwhile. For this reason, the maximum impact of the selective packet delay attack is directly related to the replay protection and to the number of packets per clock synchronization interval at the network location the attacker has access to.

Therefore, replay protection should be configured strictly (when possible) such that overtaking of packets is not allowed at all. Packets may still be delayed maliciously, however, until the subsequent packet arrives. Also, the attacker may drop or delay the subsequent packet as well in order to increase the malicious delay for the PTP packets. If there are no additional security measures in place such delay or drop of packets will not raise any suspicion. In any case, the strict replay protection in conjunction with sufficient packets per clock synchronization interval may limit the impact of selective message delays. It needs to be stressed that this mitigation depends on additional cover traffic within the entire network path from master to slave (and vice versa) which may not be under the defender's control. Furthermore, strict replay protection is not feasible in all scenarios because reducing packet loss rate may be more important in some scenarios than stricter replay protection.

Countermeasures against asymmetric OWD attacks (i.e., limiting OWDs - to be introduced in Subsection 10.1.4) are also applicable as countermeasures against selective message delay attacks (but countermeasures against selective message delay attacks are not applicable to asymmetric OWD attacks).

10.1.3 Asymmetric One-Way Delay Attacks

In Subsection 10.1.1, we introduced selective packet delay attacks, in which the attacker aims to identify PTP messages (in an encrypted traffic stream) in order to delay specific PTP packets selectively. In this subsection, we present a distinct delay attack that we denote in the following as *asymmetric OWD attack*. In such asymmetric OWD attack, the attacker does not just delay particular packets (either SYNC or DELAY_REQUEST) but delays **all** packets in one direction of the communication path but not in the other direction (e.g., all packets from the master to the slave are maliciously delayed but those from slave to master are not).

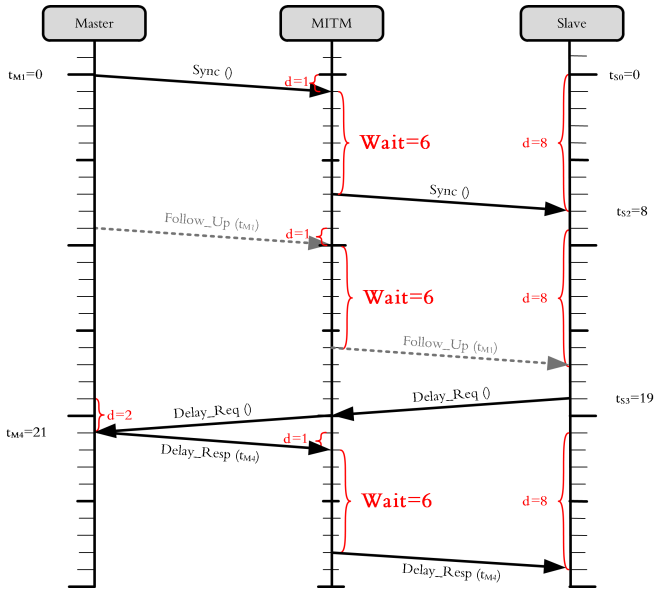

Figure 10.9: Asymmetric one-way delay attack where the master→slave delay is altered.
Adapted from [VIII].

In order to analyze how delaying all packets in one direction affects clock offset calculation we use the example discussed in the beginning of this section (with symmetric OWDs of 2 time units). This time, however, all packets sent by the master to the slave are maliciously delayed (by 6 time units) but the messages sent by the slave to the master are not, as illustrated in Fig. 10.9. We assume for simplicity reasons that clocks are neither desynchronized nor drifting. The slave miscalculates the offset according to Eq. 2.7 as 3, while the real clock offset is 0. For this reason, the slave sets its clock backwards by 3 time units. Note that the miscalculated clock offset (3) is half of the introduced delay asymmetry. If the attacker conducts the attack in reverse direction (i.e., the packets from slave to master are maliciously delayed but the packets from master to slave are not (Fig. 10.10), the slave miscalculates the offset as -3 (according to Eq. 2.7) when it is actually 0. For this reason, the slave will set its clock ahead by 3 time units.

An attacker who conducts an asymmetric OWD attack by delaying all packets in one direction can, therefore, manipulate the clock offset by half of the

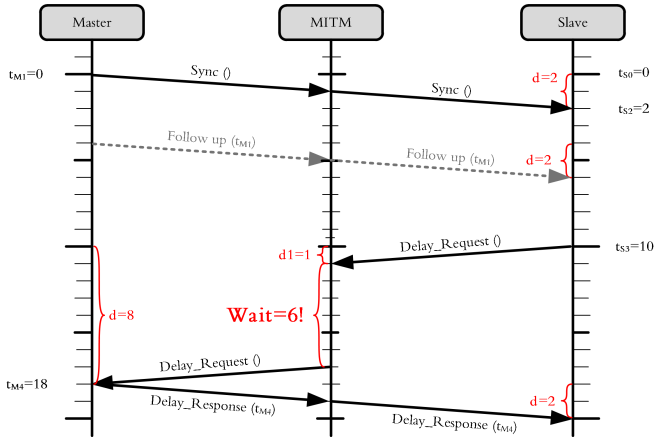

Figure 10.10: Asymmetric one-way delay attack where the slave→master delay is altered.

Adapted from [VIII].

asymmetric delay. The attacker can also influence the sign of the malicious offset correction by choosing which direction of packets are delayed maliciously. Asymmetric one-way delays being closely related to clock offset correction, two questions arise: (1) what is the exact relation between clock offset correction and asymmetric one-way delays, and (2) can asymmetric OWD attacks be prevented? These two questions will be examined in the remainder of this section.

When analyzing clock synchronization messages in detail, the first message is sent at t_{M1} by the master and received at t_{S2} by the slave. In the case of a hypothetical zero-delay path, the difference $t_{S2} - t_{M1}$ would represent the exact offset of the slave clock relative to the master. As pointed out in Section 2.8, clock synchronization messages in real systems experience various (constant and stochastic) delays along their path from master to slave and vice versa. The receiving time t_{S2} as well as the slave-computed clock offset incorporate the sum of all of those delays.

In practice, transmission and propagation delays account for the main part of the clock synchronization message end-to-end delay. This is why clock synchronization protocols comprise delay measurement methods to infer on applicable delays in order to compensate for them. On top of these measurements, high-precision clock synchronization protocols such as PTP propose dedicated functionality in intermediate devices (so-called transparent clocks in routers and switches) to compensate for queuing and processing delays within intermediate systems, even though those delays amount to a minor part of the overall delay in typical networks.

To facilitate the compensation of delays, two assumptions are required: (1) the relative clock drift within one measurement interval is negligible, and (2) the OWDs are symmetric, i.e., the sum of delays from master to slave equals the sum of the delays from slave to master. Fig. 10.11 shows the OWDs d^{MS} and d^{SM} in the general case. The offset of the slave at time t_{S2} is $offset(t_{S2}) = t_{S2} - t_{M1} - d^{MS}$ and at time t_{M4} is $offset(t_{M4}) = -t_{M4} + t_{S2} + d^{SM}$. The offset is always calculated from the perspective of the slave so that the master inverts its offset calculation (as the offset of the slave clock relative to the master is the inverse of the offset of the master clock relative to the slave). The fact that always the slave's offset is calculated can be exploited by an attacker that maliciously alters delay asymmetry, even if the attacker does neither know the

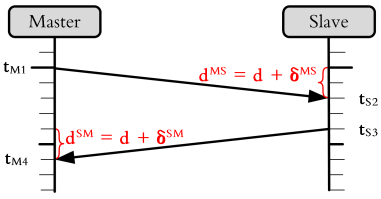

Figure 10.11: One-way delays in clock synchronization.

Adapted from [VIII].

content of a message (because it is encrypted) nor its type (because traffic is perfectly obfuscated). As we will highlight in Subsection 10.1.5, the direction alone is sufficient to manipulate the measured clock offset.

The assumptions mentioned above are essential to compensate the delays as the OWDs are approximated as half of the measured RTD. If those assumptions do not hold true, the system of two equations and four unknown variables in Eq. 10.1 cannot be solved. The first assumption (relative clock drift is negligible during one measurement interval) implies that the clock offset at time t_{S2} is (roughly) identical to the offset at time t_{M4}. For this reason, one variable is eliminated as $offset = offset(t_{S2}) = offset(t_{M4})$. The second assumption (symmetric delays) helps to eliminate another variable as $d = d^{MS} = d^{SM}$, where d^{MS} is the one-way delay from master to slave and d^{SM} is the one-way delay from slave to master as shown in Fig. 10.11. This way, there are two equations with two variables, which can be solved.

In an asymmetric OWD attack an attacker exploits the assumption on symmetric delays. Assuming that there is a common part δ in OWDs and additional two distinct delays d^{MS} and d^{SM} for both directions (as shown in Fig. 10.11), the two offset equations are as follows:

$$\begin{aligned} offset(t_{S2}) &= t_{S2} - t_{M1} - \delta - d^{MS} \\ offset(t_{M4}) &= -t_{M4} + t_{S3} + \delta + d^{SM} \end{aligned} \tag{10.1}$$

such that the offset can be calculated as

$$offset = \frac{t_{S2} - t_{M1} - t_{M4} + t_{S3} - d^{MS} + d^{SM}}{2} \tag{10.2}$$

as long as $offset(t_{S2}) \approx offset(t_{M4})$ holds true. The symmetric delay component δ is completely eliminated from the equation but d^{MS} and d^{SM} remain. If the delay is symmetric, then they cancel each other (as $d^{MS} = d^{SM}$) so that the offset can be calculated precisely. But if the delays are not symmetric, offset calculation will be off by $\frac{d^{SM}-d^{MS}}{2}$, which is in the interval $\left[\frac{-d^{MS}}{2}, \frac{d^{SM}}{2}\right]$ given that an attacker will eventually maximize either d^{MS} and keep d^{SM} close to zero or keep d^{MS} close to zero and maximize d^{SM}.

10.1.4 Countermeasures Against Asymmetric One-Way Delay Attacks

In this subsection, we propose a method that facilitates defining guaranteed bounds for the clock offset of the slave relative to the master in adversarial settings. For the following discussion we assume that master and slave clocks are synchronous at time $t_{S0} = t_{M1} = 0$ and that there is no clock drift during a clock synchronization interval. All PTP messages are cryptographically signed and encrypted, so the MITM can not modify timestamp values within these messages. The FOLLOW_UP message is omitted, and we assume that the slave immediately sends the DELAY_REQUEST after reception of the SYNC message, i.e., $t_{S3} = t_{S2}$, for sake of simplicity. However, it is important to stress that neither the initial clock synchronization nor the immediate sending of DELAY_REQUEST are a prerequisite for the proposed method. Sending DELAY_REQUEST messages at arbitrary times is essential for avoiding DELAY_REQUEST message collisions from multiple slaves following SYNC multicasts by the master, which may cause link-layer collisions in the network. However, the (in)equations below consider already separate timestamps t_{S2}

and t_{S3}, so slaves can use random delays after receiving the SYNC message in order to avoid potential collisions.

In Fig. 10.12, the master sends the SYNC message at time $t_{M1} = 0$ and receives the slave's DELAY_REQUEST at time $t_{M4} = 14$. The slave receives the master's timestamp $t_{M4} = 14$ in the DELAY_RESPONSE message and computes the RTD as 14 (according to Eq. 2.6 on page 28). The default assumption of PTP is that communication paths are symmetrical and, therefore, the timestamps $t_{S2} = t_{S3}$ must be mapped to the center of the master's interval (because the clock offset is 0), i.e. $t_{S2} = t_{S3} = 7$ as shown in Fig. 10.12.

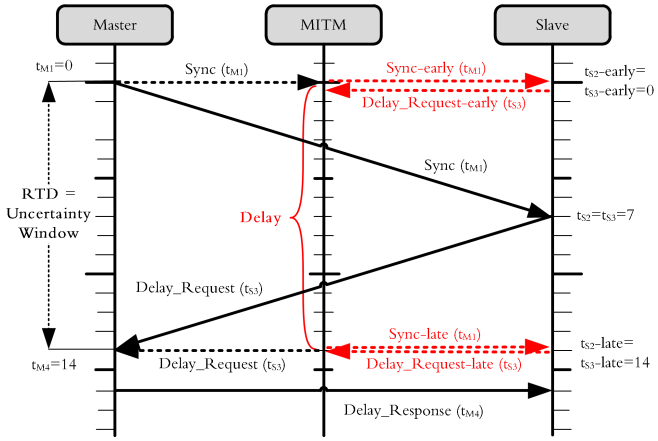

Figure 10.12: Theoretical (worst case) offset uncertainty bound calculation.

Adapted from [VIII].

Unless there is specific information available on physical delays for the forward as well as for the reverse communication path, it must be assumed that the delay of one or both of these paths can be (close to) zero. This assumption gives an adversary the opportunity to arbitrarily delay the master's SYNC message on the forward path or the slave's DELAY_REQUEST message on the reverse path within the given window of 14 time units (as depicted in Fig. 10.12). The MITM adversary can forward the master's SYNC message without additional delay (SYNC EARLY) and delay the slave's DELAY_REQUEST EARLY message by 14 time units, resulting in slave timestamps $t_{S2}\text{-}early = t_{S3}\text{-}early = 0$. Alternatively, the adversary can delay the master's SYNC message (SYNC LATE) and forward the slave's DELAY_REQUEST LATE message without additional delay, resulting in slave timestamps $t_{S2}\text{-}late = t_{S3}\text{-}late = 14$. Therefore, depending on which scenario the adversary MITM adopts, the adversary can shift the slave's offset within $[-7, 7]$ time units (according to Eq. 2.7 on page 28). Whenever the slave depends on guaranteed bounds of its clock offset in adversarial settings, this uncertainty must be considered.

Bound Clock Offset With Knowledge on OWDs

In order to reduce the uncertainty and to guarantee bounds on the clock offset, we present a method that builds upon knowledge on physical parameters and constraints of the communication path. In this way, the attacker's ability to conduct delay attacks is reduced, and the slave is supported in determining stricter guaranteed bounds for its clock offset. It is worth noting that the method can only mitigate delay attacks but not prevent them entirely.

We assume that the communication path is asymmetric and its minimum OWD is known for both directions. We denote the minimum OWD from

master → slave as d_{min}^{MS} and the minimum OWD for slave → master as d_{min}^{SM}. The exact measurement of OWDs depends on precisely synchronized clocks, which is why in real-world scenarios d_{min} may be approximated using topology and physical parameters like propagation-, transmission- and processing delays of the network path's links and components. A mandatory precondition for guaranteed clock offset bounds is that the approximated OWD must be less or equal to the minimum real packet delay on the path. Conservative approximations on d_{min}^{MS} and d_{min}^{SM}, i.e., lower minimum OWD values are detrimental to the offset bounds but are essential to guarantee the bounds in adversarial settings.

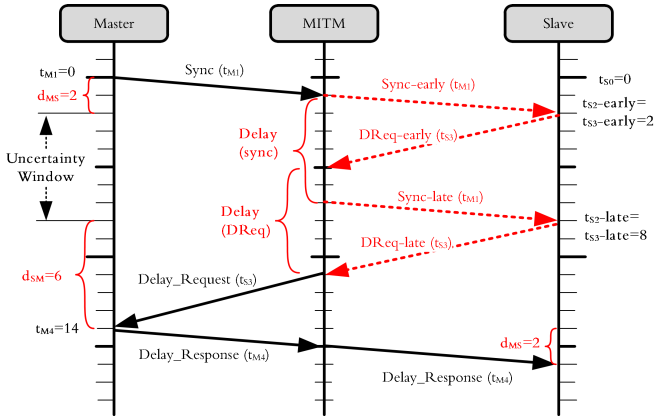

Figure 10.13: Offset uncertainty bound calculation using one-way delay limits.

Adapted from [VIII].

For computing guaranteed offset bounds using d_{min}^{MS} and d_{min}^{SM}, the slave first measures the RTD according to Eq. (2.6). In Fig. 10.13 the forward communication path from master to slave has a known minimum OWD of $d_{min}^{MS} = 2$ and the reverse path a minimum OWD of $d_{min}^{MS} = 6$ time units. Relying on the minimum delay constraints, whenever the slave receives the master's SYNC message, it knows that the master assigned timestamp t_{M1} at least $d_{min}^{MS} = 2$ time units earlier then the slave's reception timestamp t_{S2}. The slaves also knows when sending its DELAY_REQUEST message that the master will receive it and assign timestamp t_{M4} at earliest $d_{min}^{MS} = 6$ time units later than slave time t_{S3}. Using this knowledge and the timestamp t_{M4} it received in the master's DELAY_RESPONSE message, the slave can rely on the inequalities Eq. 10.3a and Eq.10.3b to hold true.

Assuming an (unknown) clock offset *offset* between the slave and master clocks, the slave can define the causal ordering of timestamps using the inequalities Eq. (10.3a) and Eq. (10.3b). All terms except the offset being known, by reordering the inequalities the slave can bound its clock offset after a clock synchronization interval by Eq. (10.3c).

$$t_{M1} + d_{min}^{MS} + \mathit{offset} \leq t_{S2} \tag{10.3a}$$

$$t_{M4} \geq t_{S3} + d_{min}^{SM} - \mathit{offset} \tag{10.3b}$$

$$t_{S3} - t_{M4} + d_{min}^{SM} \leq \mathit{offset} \leq t_{S2} - t_{M1} - d_{min}^{MS} \tag{10.3c}$$

Applying Eq. (10.3c) to the scenario in Fig. 10.13 yields bounds of $[-6, 0]$ for the early case $t_{S2} = t_{S3} = 2$ and $[0, 6]$ for the late case $t_{S2} = t_{S3} = 8$, which maps to the uncertainty window depicted in Fig. 10.13. In order to

obtain a clock offset bound that centers around 0 (despite asymmetric communication path and delay attacks), we suggest replacing clock offset calculation from Eq. (2.7) by Eq. (10.4), which basically averages the minimum and maximum offset, assuming that the uncertainty (i.e., difference between the measured RTD and the sum of the minimum OWDs d_{min}^{MS} and d_{min}^{SM}) affects the forward or reverse path with equal probability. In Fig. 10.13 the average offset is mapped to the center of the marked uncertainty window. This yields a symmetrical interval for guaranteed clock offset bounds that formally satisfy Eq. 10.5. For the scenario in Fig. 10.13, the new offset calculation results in the same size of the uncertainty window (i.e., 6) but the offset within $[-3, 3]$ centered around 0 (according to Eq. (10.5)). The knowledge of the physical delay therefore allows a stricter bound of the guaranteed clock offset compared to the case without knowledge on the OWDs presented in Fig. 10.12. In this way, the adversary's degree of freedom in manipulating clock synchronization is effectively decreased and the guaranteed bounds on the clock offset are improved.

$$\mathit{offset} = \frac{\mathit{offset}_{low} + \mathit{offset}_{high}}{2} = \frac{t_{S2} - t_{M1} - d_{min}^{MS} + t_{S3} - t_{M4} + d_{min}^{SM}}{2} \tag{10.4}$$

$$-\frac{RTD - d_{min}^{MS} - d_{min}^{SM}}{2} \leq \mathit{offset} \leq \frac{RTD - d_{min}^{MS} - d_{min}^{SM}}{2} \tag{10.5}$$

While the clock offset can be guaranteed after a clock synchronization interval (Eq. 10.5), the question about the clock offset that can be guaranteed for a particular system remains open, as an attacker could still increase the RTD arbitrarily (and therefore manipulate the clock offset). To prevent such clock offset manipulation, the RTD that is accepted needs to be restricted. Intuitively, the lower the maximum RTD that is accepted (RTD_{max}), the tighter the bound on the clock offset that can be guaranteed, but also the higher the probability of clock synchronization intervals to be discarded during unfavorable network conditions.

In order to derive clock offset bounds for a system, a time interval T_I needs to be defined that represents the maximum time interval between any two consecutive clock synchronization intervals that satisfy $RTD \leq RTD_{max}$. Between two consecutive clock synchronization intervals, the slave clock drifts at most $T_I \cdot |\rho|$, with ρ being the maximum relative clock drift of slave and master clocks. Generally, one can say that the faster and the smaller the network, the smaller RTD_{max}. Furthermore, the better the network operations, the smaller T_I, and the better the clocks, the smaller ρ. Eq. (10.6) shows the guaranteed clock offset bounds for a system.

$$\begin{aligned} -\frac{RTD_{max} - d_{min}^{MS} - d_{min}^{SM}}{2} - T_I \cdot |\rho| &\leq \mathit{offset} \\ \mathit{offset} &\leq \frac{RTD_{max} - d_{min}^{MS} - d_{min}^{SM}}{2} + T_I \cdot |\rho| \end{aligned} \tag{10.6}$$

While the relative clock offset can be bounded in adversarial settings, it needs to be stressed that high-precision clock synchronization requires significantly tighter clock synchronization guarantees than can be provided by

the bounds in Eq. 10.6. For an exemplary system we assume a small and fast network with $d_{min}^{MS} = 500\,\mu s$, $d_{min}^{SM} = 600\,\mu s$, and $RTD_{max} = 2.5\,ms$. Furthermore, we assume very good network operations with $T_I = 6\,h$ and very good master and slave clocks with $|\rho| = 2 \cdot 10^{-2}$ ppm. According to Eq. 10.6, the clock offset that can be guaranteed by such (significantly above-average quality) system in adversarial settings is $\pm$ 1132 µs, which is orders of magnitudes greater than required for high-precision clock synchronization (i.e., 100 µs down to even sub-microsecond precision as highlighted in Section 2.8). The system parameters that would be required to achieve such high-precision clock offset guarantees in adversarial settings seem completely unrealistic with current technologies[26].

[26] One could argue that deterministic networks might help as they provide guarantees on the maximum RTD and delay variation, but they depend on a precise notion of time themselves in the first place.

10.1.5 Clock Synchronization: Either Precise or Secure

The outcome of Subsection 10.1.3 that an attacker can manipulate offset correction by $\left[\frac{-d^{MS}}{2}, \frac{d^{SM}}{2}\right]$ through asymmetric one-way delays raises the question of whether a clock synchronization protocol can be designed that can handle asymmetric delay, either real existing or maliciously introduced asymmetric delay, such that its offset calculation remains unaffected in adversarial settings. We argue that constructing such protocol is impossible as delay cannot be distinguished from clock offset. The main reason is that asymmetric one-way delays can only be measured with synchronized clocks, and it can not be guaranteed that the asymmetric one-way delays are constant (especially in an adversarial setting). Synchronized clocks would be required in the first place to measure asymmetric one-way delays in order to have secure clock synchronization after all (and clock synchronization is not required when clocks are synchronized to begin with).

If we suppose an oracle to exist that can instantaneously read the clocks of master and slave, the oracle has knowledge of the real clock offset (according to Eq. 2.7) and is not influenced by asymmetric one-way delays. The clock offset measured by the slave includes asymmetric one-way delays (Eq. 10.7), however.

$$offset_{measured} = offset_{real} + \frac{d^{SM} - d^{MS}}{2} \tag{10.7}$$

If the oracle analyzes two consecutive offset measurements (at time i and $i+1$) during which clock offset was not corrected, it would observe that the difference between the offsets measured at time i and at time $i+1$ consists of two distinct parts: (1) the change in the real offset ($offset_{real_{i+1}} - offset_{real_i}$), and (2) the change in the asymmetry of one-way delays ($\frac{d_{i+1}^{SM} - d_{i+1}^{MS}}{2} - \frac{d_i^{SM} - d_i^{MS}}{2}$). The change in real clock offset (1) is a result of the relative clock drift between master and slave clocks, and the change in asymmetry of one-way delays (2) is determined by delay variation such as network jitter for example.

The relative clock drift, which determines the change of the real offset, depends on the quality of the physical oscillators used and typically ranges from 10^1 to 10^{-6} ppm. The delay variation is highly indeterministic and depends on various factors such as the current network load as well as the quality of the network and its components. Delay variation typically ranges from 10^5 to 10^2 ppm. It is important to note that delay variation is by orders of magnitudes larger than the relative clock drift, and that only an oracle could distinguish

the change of the real clock offset from the change of the asymmetry in one-way delays. For master and slave, however, they are indistinguishable as only the sum of the change can be observed in terms of measured clock offset. This means that an attacker can exploit this indistinguishability to conduct and hide an asymmetric OWD attack. As soon as a clock synchronization protocol aims to achieve high precision, it needs to entail delay measurements in order to compensate for delays. And this delay compensation mechanism is susceptible to asymmetric OWDs because OWD variation cannot be separated from clock drift.

The other issue is the direction of messages. To conduct an asymmetric OWD attack, the attacker only needs to know the direction of messages, which is tied to the master and slave roles because it is always the clock offset of the slave relative to the master that is calculated. Delaying messages in one direction has the inverse effect on clock synchronization than delaying messages in the reverse direction (as the slave clock should be synchronized to the master and not the other way around). For this reason we conclude that no clock synchronization protocol can be designed that is precise and prevents delay attacks entirely (even when messages are obfuscated in terms of length and timing and cover traffic exists)—as long as the attacker can observe the direction of the messages.

If clock synchronization protocols can be either high-precision or secure against delay attacks, then applications and in particular data origin authentication schemes must not rely on a precise notion of time when employing untrusted communication networks—if applications rely on a precise notion of time, then they must be run over trusted networks. This conclusion is (not limited to but) especially important to critical infrastructures.

10.1.6 Summary

In this section, we focused on attacks against clock synchronization protocols in the time domain, which means that protocol message content is not altered and only the timing of messages is changed. One assumption is that the attacker is in a privileged network position[27]. We first conducted a statistical traffic analysis of PTP and identified properties of PTP traffic with regard to timing, packet length, and packet directions. We showed that these properties can be used to identify PTP message types in encrypted traffic in order to conduct selective message delay attacks. Encryption schemes alone, therefore, can not provide reasonable security against selective message delay attacks.

We explored various countermeasures to mitigate selective packet delay attacks. The first set of countermeasures aims to obstruct traffic analysis. To this end, PTP can be modified in a way that randomizes the timings and the use of packet length padding, although such modification may have a negative impact on the clock synchronization's precision. Security, nevertheless, depends on the existence of suitable cover traffic, which leads to the field of traffic obfuscation. Furthermore, strict replay protection should be activated when possible to minimize the impact of the attack (and to make the attack easier to detect as the packet loss rate increases).

Then we introduced asymmetric OWD attacks. While asymmetric OWD attacks have potentially less impact on clock synchronization's precision, we

[27] While we assume the difficulty of gaining such privileged network position is within the power of an attacker who attacks critical infrastructures, we think that asymmetric OWD attacks specifically might be conducted even from non-privileged network positions by influencing the queues of network devices in a particular direction (for example by sending an excessive number of packets).

found that they are fundamentally tied to the goal of high-precision. Bounding the uncertainties of the clock offset by applying knowledge of the physical parameters of the communication path (i.e., limiting OWDs) ensures that individual messages cannot be delayed arbitrarily. Until now, network-based clock synchronization protocols asked for deterministic delays, and delays could be either symmetric or have a known asymmetry to be compensated by the PTP configuration. The results show that knowledge of the underlying communication networks is essential to limit delay attacks and to safeguard maximum guaranteed bounds on clock offset in adversarial settings. Nevertheless, asymmetric OWD attacks can only be mitigated but not be prevented entirely.

We argue that no high-precision clock synchronization protocol can exist that prevents asymmetric OWD attacks entirely because of the delay compensation mechanism that is required to achieve high-precision. In adversarial settings, an attacker can manipulate the delay variation in such a way that paths become asymmetric and clock offset calculation is impaired maliciously since clock drift and delay variation cannot be distinguished. This implies that clocks synchronization cannot be arbitrarily precise while maintaining security against delay attacks. This finding contradicts the common belief that clocks synchronization over untrusted networks can be secured by encryption and authentication methods, while improving precision. Given the results from this section, we argue the contrary: clock synchronization can either be precise or secure against delay attacks (but not both!).

Delay attacks are an inherent threat for high-precision clock synchronization since the times when messages are sent and received have an actual effect on the precision of clock synchronization and even small differences can have a large impact. The impact of those delay attacks can only be bounded but those attacks limit the precision of clock synchronization, nevertheless. Practically achievable bounds are nowhere near what some critical infrastructure applications assume today. Those infrastructures are supposed to improve specific areas but also introduce a new attack vector by their strict dependency on a precise notion of time.

10.2 SecureTime Protocol

Given the evaluation from Chapter 8, most data origin authentication schemes are unsuitable for securing multicast clock synchronization. Multicast clock synchronization requires computational efficiency and immediate signing so that clock synchronization precision is not negatively affected, low communication overhead, packet loss resistance, collusion resistance, and independence of clock synchronization. Unrestricted-time high-speed signing meets the criteria perfectly (as to be shown next), which is why we are using it in our *SecureTime* protocol[28]. To verify the suitability of unrestricted-time high-speed signing for providing data origin authentication to multicast clock synchronization, we tested the two high-speed signature schemes highlighted in Chapter 9: EdDSA and MQQ-SIG.

[28] We would like to note that SecureTime does not depend on a particular data origin authentication scheme, nonetheless.

10.2.1 Measuring the Impact of Signing on Clock Synchronization Precision

To test whether unrestricted-time high-speed signing actually delivers the expected performance in practice we conducted experimental measurements us-

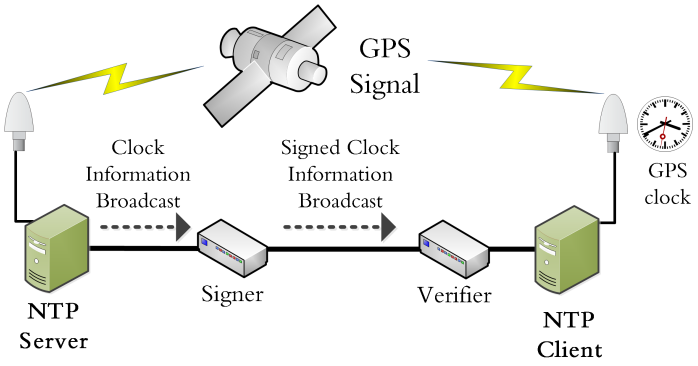

Figure 10.14: Measurement of signed NTP messages.
Adapted from [III].

ing the measurement setup depicted in Figure 10.14. We set up a stratum 1 NTP server that is synchronized to UTC through a dedicated GPS PPS hardware clock. The server broadcasts clock synchronization messages to the local network every 8 seconds, which is the shortest interval possible without making changes to NTP's source code. On the other end runs an NTP client that synchronizes its clock to the NTP server's broadcast time. To measure clock synchronization precision, the NTP client can access the same GPS PPS timing signal as the server, but is configured to synchronize its clock to the NTP server and to explicitly ignore the GPS PPS signal in its clock synchronization algorithm; i.e., the client just logs the GPS PPS values to support later assessment of the influence that unrestricted-time high-speed signing has onto NTP's precision. Both end systems, NTP server and NTP client, use identical hardware for synchronization and timestamping.

Between the server and the client are two network bridges: *Signer* and *Verifier*. Signer waits for NTP packets and signs the payload using a high-speed digital signature scheme. Verifier looks for signed NTP packets, verifies the signature and removes it thereafter. At the client normal NTP messages arrive. The described setup allows running unmodified NTP code. Signing messages and verifying signatures takes time, so that messages arrive delayed at the slave, and therefore affect clock synchronization precision. The GPS PPS synchronized reference clock at the client allows quantifying the influence on clock synchronization precision.

We assume the influence on precision to be small due to the use of high-speed signature schemes. To separate the influence introduced by our measurement setup from the influence introduced by signing and verifying messages, we conducted four measurements. Each measurement was run for more than 24 hours.

First, we configured Signer and Verifier as plain network bridges that only forward packets. In this scenario, we observed a median clock offset of 241 µs, which is the influence that Ethernet plus two bridges have on NTP's precision (when the delay is not compensated). Then, Signer and Verifier not only forwarded packets but first copied the packets to user space, where they were parsed (but not signed!). This introduced a median offset of 336 µs, which includes the offset for Ethernet as measured before. We therefore conclude that copying the packets to user space and parsing them (two times) introduces around 94 µs of additional offset. The last two measurements finally quantify the offset that is added to NTP's clock synchronization algorithm by

signing and verifying each packet individually using the signature schemes under test: Ed25519 and MQQ-SIG. For Ed25519, we observed a median offset of 411 µs, which means that about 75 µs were spent on signing messages and verifying signatures. For MQQ-SIG, we observed a median offset of 364 µs. This means that only about 28 µs were spent on signing messages and verifying signatures. Table 10.4 summarizes the measurement results, which are in line with and therefore confirm the previous results (from Chapter 9).

Measurement	Mean Clock Offset	Median Clock Offset	Std. dev.	Relative Clock Offset
Ethernet	239 µs	241 µs	5.5 µs	−94 µs
Ethernet + nfqueue	335 µs	336 µs	8.7 µs	0 µs
Ethernet + nfqueue + Ed25519	410 µs	411 µs	8.2 µs	75 µs
Ethernet + nfqueue + MQQ-SIG	364 µs	364 µs	8.2 µs	28 µs

Table 10.4: Measurement results: NTP's clock synchronization precision.
Adapted from [III].

The actual offset introduced by signature generation and verification highly depends on the particular computer architecture and resources. Nevertheless, we argue that, when implemented within a clock synchronization protocol, the offset introduced by the authentication scheme can be further reduced as offsets measured are the sum of the offset for signing and for verification. The offset for verification can easily be compensated when implemented with a clock synchronization protocol (and not in intermediary network devices as in our experiment). For generation, the slaves could only compensate for the minimum time it takes the master to generate a signature in one-step mode (again when implemented in the clock synchronization protocol) so that only the variation remains and influences the precision. It is worth noting that our experimental measurements use a nfqueue-based implementation, which passes the packet data to user space for signing. Implementing signature generation in the kernel or in hardware would substantially decrease both the time and the variation of signature generation. Moreover, SecureTime in combination with the 2-step clock synchronization mode renders the influence on precision to practically zero (as will be explained in the subsequent subsection).

The communication overhead is low because of the small signature sizes of both schemes (32 B for MQQ-SIG and 64 B for Ed25519). For the overall overhead, the size of the sequence number has to be included per message as well as the size of the public key per session. The size of the public key is significantly larger for MQQ-SIG than for Ed25519 (32 kB vs. 517 B). The communication overhead of SecureTime in 2-step mode (discussed next) is increased because of the (signed) FOLLOW_UP message. This overhead may be well justified because of the increased clock synchronization's precision, however.

10.2.2 SecureTime's Security Measures

Since delay attacks have been covered in detail in Section 10.1 and unrestricted-time high-speed signing has been shown to be suitable (Section 10.2.1), we

Name	Data Origin Authentication	NTP Support	PTP Support	Substitution Attack Prevention	Pre-play Attack Prevention	Replay Attack Prevention	Delay Attack Mitigation
NTS	+	+	--	+	+	+	--
PTP extension	+	--	+	+	--	+	~
SecureTime	+	+	+	+	+	+	+

Property is either satisfied (+), somewhat satisfied (~), or not satisfied (--).

Table 10.5: Comparison of SecureTime to related work.

Adapted from [III].

can now focus on a complete set of security measures to secure multicast clock synchronization. In this subsection, we introduce the SecureTime protocol, a set of secure measures for securing (multicast) clock synchronization. In contrast to existing work (i.e., NTS[29,30,31] and a recently suggested security extension to PTP[32,33]), SecureTime supports both NTP and PTP. The security provided by SecureTime against the attacks identified in Chapter 4 will be analyzed as well. Furthermore, SecureTime entails detailed delay attack mitigation (and is not vulnerable to pre-play attacks). Table 10.5 summarizes the comparison of related work to SecureTime.

Freshness of Messages Replay attacks can be prevented, as long as it can be ensured that once a message has been received it is rejected when received again later; i.e., freshness of messages is provided. To this end, a data origin authentication scheme alone does not suffice, because successful verification of a message does not imply that the message has never been received before. To ensure freshness of messages in SecureTime, the master includes a sequence number in each message that is monotonically increasing, and the slave checks for increasing sequence numbers. In SecureTime, the sequence number starts with zero[34] for the first message and increases by one with every message. Slaves and the master need to store only one sequence number as state information, and it is safe and straightforward to restart a session when master or slave are not entirely sure about the current state (after restoring from a backup for example). For every received message (that has a valid signature) the slave checks the freshness of the message by verifying that the included sequence number is greater than the one locally stored. If this is the case, the message is accepted and the local state updated accordingly; otherwise, the message is discarded.

SecureTime's strict handling of sequence numbers allows for lost messages due to transmission errors but prevents a message from getting accepted by the receiver if that message was overtaken (by another message) during transmission. While this may sound like a significant disadvantage at the first glance, it is actually beneficial in the context of clock synchronization. Messages that have been overtaken in transit entail a larger delay than other messages. Since delay variation has a negative effect on the slave's notion of time (as analyzed in Section 10.1), it is beneficial to the clock synchronization's precision when those (delayed) messages get discarded.

[29] Sibold, Roettger, and Teichel, *Network Time Security*.

[30] Sibold, Roettger, and Teichel, *Using the Network Time Security Specification to Secure the Network Time Protocol*.

[31] Sibold et al., *Protecting Network Time Security Messages with the Cryptographic Message Syntax (CMS)*.

[32] Itkin and Wool, "A security analysis and revised security extension for the precision time protocol".

[33] Itkin and Wool, "A Security Analysis and Revised Security Extension for the Precision Time Protocol".

[34] There is no need for a randomly initialized sequence number for two reasons: (1) the sequence number is part of the clock synchronization messages, which is cryptographically signed so that an adversary cannot modify it unnoticedly. And (2), an adversary can barely gain information about the start of the communication from the sequence number since the adversary cannot know how often the sequence number has overflowed.

Session Keys Since clock synchronization is a continuous process, the fixed-size sequence number will overflow eventually. For this reason, a long-term attacker could capture all messages and start replaying them as soon as the sequence number has overflowed. To prevent such long-term attack, the master signs a session public key (with the long-term secret key) as well as the current sequence number and sends both to the slave at the start of the communication. The slave verifies the correctness with the master's long-term public key. In this way, also transient slaves such as laptops and mobile phones can be handled.

Before the sequence number overflows, a new session has to be started with a fresh session key pair and the sequence number is reset to zero. This introduction of session keys, used for only a limited number of messages, provides not only security against replay attacks by long-term attackers but also reduces the time an adversary has to attack a particular key pair. In this way, session keys also reduce the pressure on the employed data origin authentication scheme.

Selecting a particular session length involves a trade-off between communication (and computational) overhead and reduced pressure on the data origin authentication scheme (i.e., increased security). A shorter session length increases the computational overhead slightly as new keys need to be generated more often. On the other hand, a shorter session length also reduces communication overhead slightly because of the smaller sequence number field (although new keys also need to be distributed more often). Furthermore, the pressure on the underlying data origin scheme is slightly reduced as session keys are valid for a shorter time period.

Dynamic Clock Offset Correction Limit In SecureTime, the maximum clock offset correction that a slave applies in one synchronization interval is limited to $|t_{arr} - t_{last}| \cdot \rho$. The dynamic clock offset correction limit restricts the maximum influence that individual clock synchronization intervals can have on the slave's notion of time. This limit implies that the smaller the synchronization interval, the tighter the dynamic clock offset correction limit, and therefore the smaller the changes that are applied to slave clocks. For this reason only incremental delay attacks can be conducted, since clock synchronization intervals that are influenced by too large delays are discarded.

Slave-Specific Maximum Synchronization Interval To ensure that the clock offset between master and slave cannot be changed arbitrarily, SecureTime employs a slave-specific maximum synchronization interval T_I. Network operation engineers need to ensure that the interval between any two successful clock synchronization intervals is at most T_I. This interval is required to bound delay attacks, as shown in Section 10.1.

Secure Delay Measurement Procedure The secure delay measurement procedure consists of two authenticated unicast messages (DELAY_REQUEST and DELAY_RESPONSE). To ensure that clock synchronization messages have not been artificially delayed, the checks and improved offset calculation from Subsection 10.1.4 are employed. In the DELAY_RESPONSE message, the master furthermore includes the current sequence number as well as an identifier of

the session's public key. The slave updates its sequence number state to the number entailed in the DELAY_RESPONSE message and sets t_{last} to the time of when it corrected its clock. Furthermore, the slave checks that the session public key identifier matches the session public key (if it does not match, the slave needs to restart the communication). For this reason, an adversary cannot prevent the transition to a new session arbitrarily, and the time an adversary can attack an old key is bounded.

10.2.3 Security Analysis of SecureTime

We assume that the adversary is in a privileged network position and conducts various, potentially severe attacks (as described in Chapter 4 and Section 5.1). We furthermore assume that the adversary does not know any secret key of the master, but may know the master's public key. With respect to the data origin authentication scheme, we assume that it provides existential unforgeability.

We show in this subsection that an adversary cannot make a slave adhere to a false time by substitution attacks, impersonation attacks, or replay attacks when the slave receives clock synchronization messages from a honest master and both slave and master employ the SecureTime protocol. Furthermore, we will show that the maximum impact of delay attacks is bounded.

Substitution Attack In a substitution attack, the adversary first intercepts messages including the corresponding authentication information. Then, the adversary substitutes parts of an intercepted message such that the slave is falsely convinced that the modified message originated from the master. We argue that a substitution attack can only be conducted successfully when breaking the data origin authentication scheme. For substitution attacks, we distinguish two cases: (1) the adversary modifies the message in such a way that the resulting message is identical to a message the adversary has intercepted. For the intercepted message, the adversary also intercepted the corresponding (valid) signature. However, this case is equivalent to either a replay attack or to a delay attack, depending on whether the adversary dropped the original message. The security against replay attacks as well as against delay attacks is analyzed separately later. (2) The adversary modifies the message in such a way that it is new. In order to make a slave accept the new message, the adversary needs to provide authentication information that is valid under the master's session public key. If the adversary can generate valid authentication information with non-negligible probability he could also efficiently forge signatures, which contradicts the existential unforgeability provided by the data origin authentication scheme.

Pre-play Attack In order to get a pre-played message accepted by a slave, the adversary needs to generate a valid signature for that message. Again, if the adversary can do that efficiently he could also break the existential unforgeability of the underlying data origin authentication scheme.

Replay Attack In a replay attack, the adversary injects a message that was intercepted before and therefore includes valid authentication information provided by the master. We argue that a slave that employs SecureTime discards

replayed messages unless the adversary breaks the underlying data origin authentication scheme. To this end, we distinguish two cases: (1) the adversary prevented the original message from reaching the slave, which is equivalent to a delay attack where the adversary holds a message for some time and forwards it later. Delay attacks, specifically, are analyzed later. (2) The adversary did not prevent the original message from reaching the slave. Since the slave has received the original message, the slave updated its local state with the sequence number from the received message. If no other message reached the slave in the meantime, the sequence number in the replayed message is identical to the slave's local state. For this reason, the slave will discard the replayed message. When other messages have reached the slave in the meantime, we need to distinguish two sub-cases: (2*a*) the maximum sequence number value was not reached. In this case, the sequence number in the replayed message is smaller than the slave's local state, and the slave will therefore discard the replayed message. (2*b*) The maximum sequence number value was reached. In this case, the sequence number of the replayed message may actually be greater than the sequence number last seen by the slave at that time because the sequence number was reset to zero when the new session was started. The authentication information of the replayed message, however, is not valid anymore since the master has switched to a new key pair for the new session. The adversary, therefore, needs to generate authentication information that is valid under the new key pair in order to make the slave accept the message. This is equivalent to an substitution attack, however, which was analyzed previously.

Delay Attack In a delay attack, the adversary intercepts a message, delays it for some time before forwarding it (as described in detail in Section 10.1. In contrast to the other attacks, delay attacks cannot be prevented entirely by data origin authentication. We have shown, however, that delay attacks can be mitigated by providing a worst case bound to the clock offset between master and slave clocks (Eq. 10.6).

10.2.4 Extending SecureTime to Two-Step Mode

Basically all previously introduced security measures for the one-step mode can directly be applied to the two-step mode. The two-step mode has an appealing property, however, that allows SecureTime to have practically zero influence on the clock synchronization's precision. In two-step mode, there are two messages for clock offset correction, a SYNC like in one-step mode and an additional FOLLOW_UP message. For Securetime in two-step mode, not only the SYNC message is signed but also the FOLLOW_UP message. While this sounds counter-intuitive at the first glance (why should have signing two messages have practically no impact on the clock synchronization's precision when signing one message has?), we find this solution quite elegant.

To secure two-step mode, SecureTime makes use of the hardware timestamping feature of NICs that is provided by most recent NICs[35]. With hardware-timestamping, the NIC recognizes SYNC messages and stores the point in time when the SYNC messages was put on the wire. The SYNC message itself has no meaningful content - it solely serves as timestamp. Before sending the FOLLOW_UP message, the NIC is queried for the time when the

[35] In case the particular NIC does not support hardware-timestamping, software-emulation can be used, which has slightly worse precision.

SYNC message was put on the wire and that time is inserted in the FOLLOW_UP message. Since the specific time when the FOLLOW_UP message is sent or received is of no particular importance, the time it takes to sign a message or to verify its signature is not of importance either. Therefore, signing and verifying has no influence on the clock synchronization's precision in two-step mode.

The sequence number assures that SYNC and FOLLOW_UP messages are indeed related. For this purpose, receivers additionally need to make sure that the difference of sequence numbers in SYNC and FOLLOW_UP messages is exactly 1. If this is not the case, the messages must be considered invalid. The slave can immediately discard SYNC messages that have a smaller sequence number than the last seen sequence number. But the slave must not update its local state until a valid FOLLOW_UP message was received. The slave clears its list of buffered SYNC messages after receiving a (valid) FOLLOW_UP message, after a delay measurement, and when a new session is started. Basically, FOLLOW_UP messages, delay measurements, and sessions serve as timeout periods for received SYNC messages. When receiving a FOLLOW_UP message, the slave first checks the validity of the included signature. The slave also checks that the sequence number of the SYNC message is exactly one below the FOLLOW_UP's sequence number. If this is the case, the SYNC message and also the FOLLOW_UP message are accepted as valid; otherwise, they are discarded.

Support for Transparent Clocks As pointed out earlier, transparent clocks are a significant security challenge. Nevertheless, SecureTime supports transparent clocks in a sense that they can append the offset they introduced to the original SYNC and DELAY_REQUEST messages and to the corresponding FOLLOW_UP message themselves by appending their own (signed) information (to the FOLLOW_UP and DELAY_RESPONSE messages). It cannot be guaranteed, however, that a specific transparent clock is on the path from master to slave as an adversary could just have compromised the transparent clock and use its secret key to add malicious offset. Nevertheless, this kind of attack can be seen as a special case of a delay attack, which was already analyzed.

10.2.5 Drawbacks

The SecureTime security measures comprise two drawbacks. (1) Hardware-timestamping cannot be used in one-step mode (because of signing). Since hardware-timestamping is used to improve precision and still works in two-step mode with SecureTime, we argue to use the two-step mode to begin with when high-precision is of importance. (2) Signing SYNC messages violates the PTP standard a bit because SYNC messages should not contain a Type Length Value (TLV) in order to have constant transmission delays for all messages. We argue that constant transmission delays can still be achieved as long as the DELAY_REQUEST messages are of the same size. For this reason, we conclude that both drawbacks can be handled and are therefore negligible.

11
Group Communication in 5G Networks

This chapter has been published in part in:

Robert Annessi, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252

5G NETWORKS offer new communication possibilities for critical infrastructures. In this chapter, we present two attack scenarios to group communication services in 5G cellular networks, which are facilitated because only group authentication and not data origin authentication is provided by the current group communication specific security measures for 5G. We focus on the major threat of impersonating the group communication service through injection or modification of datagrams although less severe DoS attacks can be equally conducted by the adversary[1]. Fig. 11.1 on the next page and 11.2 on page 108 depict the attack scenarios. For both attack scenarios, we assume that the adversary knows the shared group key (MTK or RNTI for MBMS and SC-PTM, respectively) and has access to a particular part of the network.

For the short-range attack scenario (Fig. 11.1), the adversary can get knowledge of the (shared) group key through a collaborating UE, by operating a legitimate UE, or by compromising a UE. The assumption that the adversary has knowledge of the group key is reasonable especially for group communication because the more UEs exist the more likely it is that at least one gets compromised, collaborates with, or is operated by the adversary. In this specific example (Fig. 11.1), it is UE_3 that (deliberately or not) cooperates with the adversary but it could equally be any other UE that is subscribed to the group communication service. The adversary needs additional access to the particular part of the network. Access to the network is a reasonable assumption as well because the (collaborating or compromised) UE already has access to the air interface. Knowledge of the shared group key and access to the network are sufficient to inject arbitrary data maliciously, which will be received by all UEs in range as authentic since the adversary can generate valid authentication information with the correct group key[2]. The impact of the attack highly depends on the specific group communication service and the content of the injected data and can be anywhere from inconvenient to catastrophic. Unless the attack is concerted in a distributed fashion on multiple locations,

[1] The adversary can, for example, inject a single datagram (that is considered authentic by the receivers) that includes the maximum possible sequence number such that UEs will discard any other future datagrams for that service from the legitimate source.

[2] It needs to be noted that network access is only a prerequisite, injecting data in a real-world scenario require more effort.

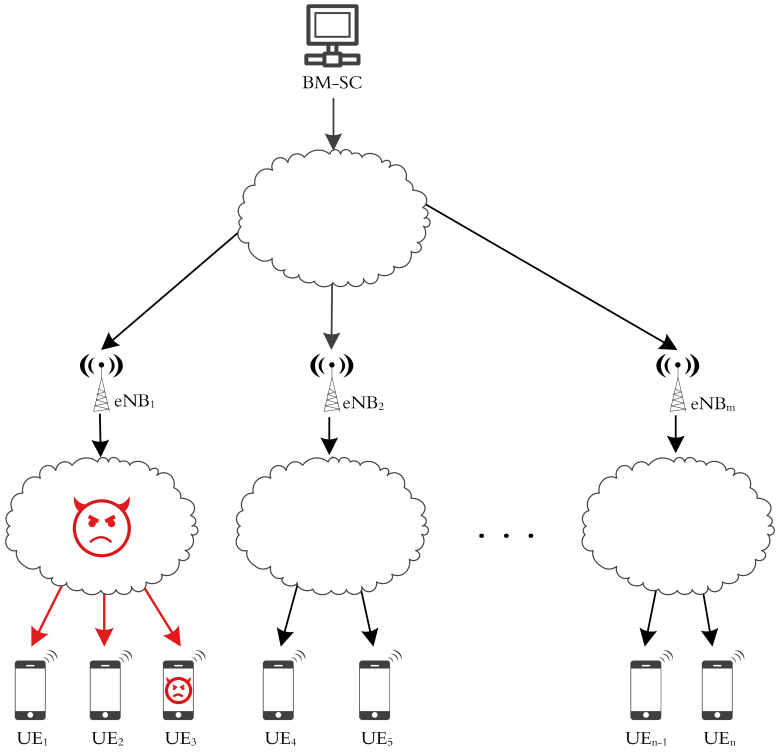

Figure 11.1: Short-range attack scenario on group communication services in 5G.

Adapted from [IV]. © ACM 2018.

the short-range attack is physically limited to the UEs within range that are subscribed to the same eNB.

In the long-range attack scenario (Fig. 11.2 on the next page), the adversary needs access to the communication network between the BM-SC and the eNBs (instead of access to the air interface needed in the first attack). The communication network may be accessed through an individual network access that was not properly secured or through a compromised network device, for example. This time, however, the adversary needs to break additional security measures. There may be three types of additional security measures: (1) there may not exist additional security at all if unicast communication is used between the BM-SC and the eNBs and all operate in the same network. In this case all preconditions are fulfilled, meaning that the attacker does not need to circumvent any additional security measures to gain access. (2) If group communication is used between the BM-SC and the eNBs, they use the group communication specific security measures (as described in Section 5.2). For this reason, the security measures again fall short of providing a sufficient level of authentication if the attacker gets access to the group key shared between the BM-SC and the eNBs. In this case, the attacker needs to compromise an eNB, which is supposedly significantly harder than compromising (or operating) a UE but, on the other hand, the attack also has significantly more impact, which may justify the additional effort. (3) The communication between BM-SC and eNB consists of multiple unicast connections that are secured with IPsec. Then, the attacker would need to break IPsec, which is commonly considered unfeasible. For this reason, mandatory IPsec-secured unicast connections between BM-SC and eNBs can be considered an interim solution. The downside, however, is that the communication between BM-SC and eNBs is rather inefficient then in terms of computational

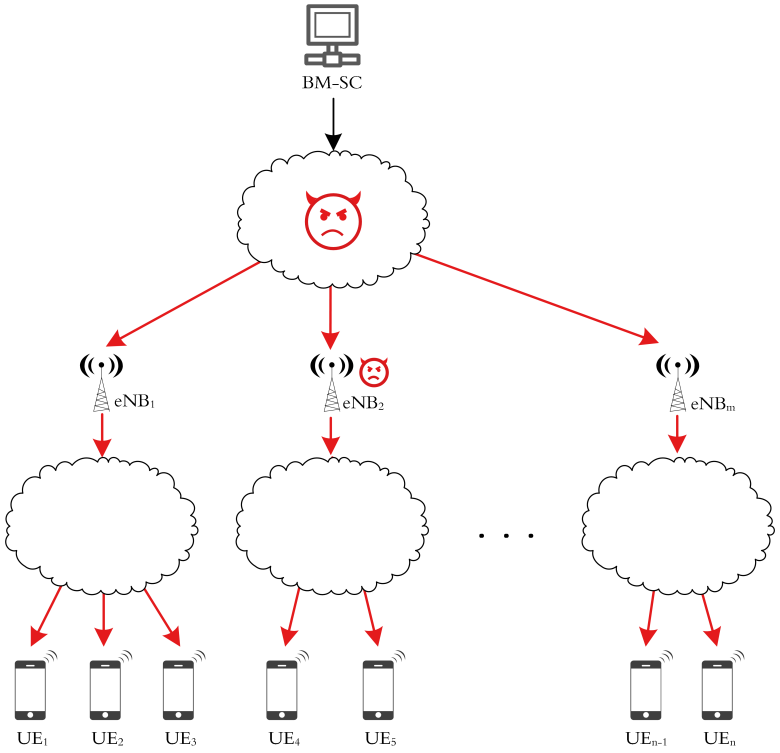

Figure 11.2: Long-range attack scenario on group communication services in 5G.

Adapted from [IV]. © ACM 2018.

and communication overhead. In any case, the impact of the long-range attack is significantly larger than that of the short-range attack scenario since the data are propagated from the BM-SC via the eNBs to UEs, and therefore every UE that is subscribed to the service receives the maliciously injected data (and considers them authentic).

To counter both attack scenarios, conventional security measures are apparently inadequate. Data origin authentication is required in order to allow the BM-SC to secure group communication without giving the UEs means to construct valid authentication information on behalf of the BM-SC. Each UE can then verify the authenticity of the data without having the necessity to trust the benignity as well as the operational security of the network operators and its users. Given the trend on attacks on critical infrastructures and its users, we argue that trusting users and networking operators is no longer appropriate nor acceptable today.

Data Origin Authentication for 5G

Given the shortcomings of group authentication, we discuss the potential of each class of data origin authentication schemes to secure group communication in 5G networks. To be suitable for securing group communication in 5G, a data origin authentication scheme needs to be computationally efficient (especially since verification may be conducted by battery-powered devices), resistant to packet loss since packet loss is inherent to wireless networks, and have low communication overhead. Based on the evaluation of data origin authentication schemes (Chapter 8), we can say that schemes from the secret-information asymmetry class require substantial computational resources for verification (and for signing) and introduce significant communication over-

head, which makes them unsuitable for 5G applications that involve low-power devices. k-time signing from the deferred signing class, on the other hand, could suite the group communication scenario in 5G if the computational resources at the UEs should be reduced even at the expense of increased communication overhead. Schemes from the signature propagation class rely on the successful reception of signature datagrams and are, therefore, hardly resistant to the loss of datagrams, which makes this class of schemes unsuitable for group communication services in 5G networks. While the signature dispersal class could be somewhat suitable to secure group communication services in 5G networks, schemes from the MTS and the unrestricted-time high-speed signing classes include more promising data origin authentication schemes. Since clock synchronization is already part of cellular networks, schemes from the MTS class may be well suitable for group communication in 5G networks—clock synchronization precision becomes a security requirement then, however. A data origin authentication scheme from the unrestricted-time high-speed signing class therefore seems most promising to secure group communication in 5G networks.

12

Sensor Data Collection in Smart Grids

One specific example of a critical infrastructure is the future power grid, commonly referred to as Smart Grid, in which power, communication, and information technologies are integrated for an improved electrical power infrastructure[1]. Smart Grids enhance electrical grids with information technology to optimize grid operation. To provide exact control and in-time anomaly detection in Smart Grids, data must be relayed fast with minimal delay over long distances to multiple receivers. In the context of Smart Grids, group communication is envisioned to be applicable to various scenarios like Wide Area Monitoring Systems (WAMSs), WAMPACSs, demand response, and Smart Energy Hubs.

Sensor data from PMUs are used to measure the phasors of electric current and voltage at various locations in Smart Grids. Up to 120 measurements are performed per second and PMU. The measurements are transmitted in real-time to various receivers such as PDCs or control centers (see Fig. 12.1). Applications based on sensor data collection such as visualization applications or event classifiers that support control decisions will likely lead to Cyber Critical Asset (CCA) status for PMUs[2]. Given the fact that Smart Grids have to be considered critical infrastructure, essential for modern society, and that decisions are made based on sensor data, ensuring the integrity and the authenticity of communication is of utmost importance.

[1] "IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads".

[2] Morris, Pan, and Adhikari, "Cyber security recommendations for wide area monitoring, protection, and control systems".

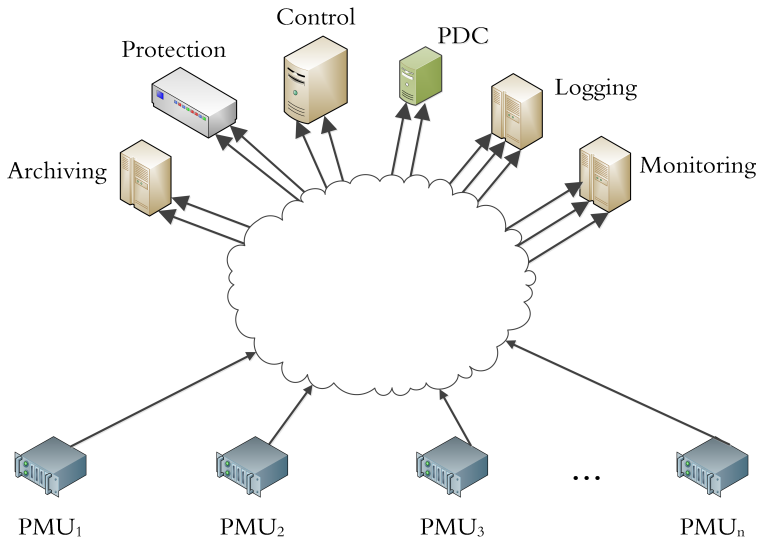

Figure 12.1: PMU sensor data multicasted to various applications in a Smart Grid.

For data origin authentication schemes that secure sensor data specifically, computational efficiency is important since sensor data may be time-critical (depending on the application they serve). For example, critical applications in Smart Grids require an end-to-end delay of at most 5 ms[3]. For this reason, the time required to generate and verify authentication information should be minimal. Furthermore, sender- and receiver-side buffering must be avoided so that packets can be signed and signatures verified immediately, respectively. Unrestricted-time high-speed signing satisfies these requirements and are therefore suitable to protect data sent from sensors to data collectors in Smart Grids.

[3] IEC/TR 61850-90-1:2010, *Communication networks and systems for power utility automation – Part 90-1: Use of IEC 61850 for the communication between substations*.

13
Subliminal Channels

This chapter has been published in part in:

Davor Frkat, Robert Annessi, and Tanja Zseby. "ChainChannels: Private Botnet Communication Over Public Blockchains". In: *IEEE International Conference on Blockchain (Blockchain)*. 2018. Referred to as "[V]". © IEEE 2018.

Alexander Hartl, Robert Annessi, and Tanja Zseby. "A Subliminal Channel in EdDSA: Information Leakage with High-Speed Signatures". In: *International Workshop on Managing Insider Security Threats*. MIST '17. Dallas, Texas, USA: ACM, 2017, pp. 67–78. ISBN: 978-1-4503-5177-5. DOI: 10.1145/3139923.3139925. Referred to as "[VI]". © ACM 2017.

Alexander Hartl, Robert Annessi, and Tanja Zseby. "Subliminal Channels in High-Speed Signatures". In: *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)* 9.1 (Mar. 2018), pp. 30–53. Referred to as "[VII]". © JoWUA 2018.

A TYPICAL USE OF SUBLIMINAL CHANNELS in network protocols are scenarios where the encrypt-then-sign paradigm is employed[1] or where encryption is not permitted, unusual, or just unintended, but the use of signatures is allowed to ensure authenticity and integrity of messages. Scenarios are particularly susceptible to subliminal channels if authenticity, integrity, or non-repudiation has to be guaranteed but confidentiality is not as important (or even undesired) such that data are sent unencrypted. This applies, for example, to group communication scenarios like multicast clock synchronization, sensor data collection in Smart Grids, or blockchains. Subliminal channels can be used in such scenarios for information leakage or malware communication for example. In this way, subliminal channels are a serious threat to information security.

Subliminal channels did not receive much attention first and were tolerated (or possibly even ignored) by protocol designers. In classical security protocols, such as IPsec or TLS, signatures are usually only used sparsely, mainly in the authentication phase. For this reason, the bandwidth for data leakage is limited to a few bytes per connection. It may still be enough to leak critical information such as secret keying material or status information (despite limited bandwidth). The advent of unrestricted-time high-speed signing coupled with newly emerged application areas changes this bandwidth limitation, however, as many packets are signed individually. This results in a significant increase of bandwidth available to subliminal channels.

[1] With the encrypt-then-sign paradigm the ciphertext is signed and the signature itself is unencrypted such that both the ciphertext and the signature are visible (to an eavesdropper) in the network.

In this chapter, we show how subliminal channels can be created using EdDSA and MQ signatures. We describe how subliminal channels can be created using EdDSA (Section 13.1), how the channels can be used in different scenarios (Section 13.2) and what methods can be used to prevent the subliminal communication (Section 13.4). We additionally performed experiments to prove the subliminal channel for various scenarios and analyze the subliminal bandwidth and the difficulty of exploiting the channel in practice (Section 13.3). We then describe how subliminal channels can be created for MQ signatures (Section 13.5). We show general possibilities for hiding information in the mathematical constructs of MQ signatures and several of their modifiers. Then we investigate the achievable subliminal bandwidth for several existing MQ signatures such as QUARTZ[2], Gui-127[3], SFlash[4], PFlash[5], MQQ-SIG[6], and Rainbow[7]. Finally, we present *ChainChannels* (Section 13.6), a novel way of secretly multicasting information over public blockchains, by embedding subliminal information in the signatures used to secure blockchain transactions.

13.1 Subliminal Channels in EdDSA

Simmons introduced a classification of subliminal channels according to the bandwidth of subliminal information[8]. For a *broadband* subliminal channel, the information can use almost the bits of a signature that are not needed for its security against forgery, and for a *narrowband* subliminal channel, the subliminal bandwidth is significantly smaller (i.e., a few bits per signature). EdDSA yields a broadband subliminal channel as well as a narrowband channel.

13.1.1 The Broadband Channel

Like in other signature schemes that are based on the discrete logarithm problem, the (cryptographically secure) nonce r can be calculated easily from a valid signature if the signing key a is known as

$$r = S - H(R, A, M)a \bmod L \tag{13.1}$$

It is noteworthy that the calculation rule for the nonce r in Eq. (2.2) only serves as a high-quality random number generator for the signature scheme. Using a different value for the nonce does not harm the successful verifiability of the produced signature in any way. The value of the nonce can therefore be used as a subliminal channel by encoding subliminal information into it. Subliminal information directly encoded into the nonce can be recovered using Eq. (13.1) by anyone who holds the signing key a and is able to intercept the message and its signature (see Fig. 2.8). Since information is encoded in the residue class modulo L, the subliminal channel has a bandwidth of at most $\log_2 L$ bits per signature. For Ed25519 this corresponds to a bandwidth of 252 bit per signature and for Ed448 to 447 bit per signature.

As mentioned, it is required for the subliminal receiver to know the signing key a. To this end, we distinguish two cases: (1) the legitimate sender wants to transmit subliminal information intentionally and (2) the legitimate sender has been compromised and the subliminal message is inserted by malware that has access to the signing process. For case (1), we assume that the sender of the subliminal information and the subliminal receiver collaborate and that it

[2] Nicolas Courtois, Louis Goubin, and Jacques Patarin. "Quartz, an asymmetric signature scheme for short signatures on PC Primitive specification and supporting documentation". In: (2001).

[3] Albrecht Petzoldt et al. "Design Principles for HFEv- Based Multivariate Signature Schemes". In: *Advances in Cryptology – ASIACRYPT*. Springer, 2015, pp. 311–334. ISBN: 978-3-662-48797-6.

[4] Nicolas T Courtois, Louis Goubin, and Jacques Patarin. "SFLASHv3, a fast asymmetric signature scheme." In: *IACR Cryptology ePrint Archive* 2003 (2003), p. 211.

[5] Ming-Shing Chen, Bo-Yin Yang, and Daniel Smith-Tone. "PFLASH - Secure Asymmetric Signatures on Smart Cards". In: *Lightweight Cryptographic Workshop*. 2015.

[6] Gligoroski et al., "MQQ-SIG".

[7] Jintai Ding and Dieter Schmidt. "Rainbow, a New Multivariable Polynomial Signature Scheme". In: *International Conference on Applied Cryptography and Network Security, ACNS 2005*. Springer, 2005, pp. 164–175. ISBN: 978-3-540-31542-1.

Attribution to paper coauthors

The work on subliminal channels in EdDSA (described in the joint papers [VI], [VII], and Sections 13.1, 13.2, 13.3, 13.4, and 13.5) was conducted in collaboration with Alexander Hartl. The practical experiments (described in Section 13.3) and the work on subliminal channels in MQ-based signature schemes (described in Section 13.5) were conducted primarily by Alexander Hartl.

[8] Simmons, "Subliminal Communication Is Easy Using the DSA".

is therefore reasonable to assume that they share the signing key a before the subliminal communication starts. With knowledge of the signing key, the subliminal receiver is equally able to generate signatures (on behalf of the sender), and this method should therefore only be used if the subliminal receiver is trusted by the sender. For case (2), the adversary needs to clandestinely leak the signing key to the receiver. For the subliminal communication scenarios described in Section 13.2 the key could be leaked using the narrowband channel described below.

13.1.2 A Narrowband Channel

A general approach for creating a narrowband subliminal channel by exploiting signature schemes that allow multiple valid signatures for a message was proposed by Simmons[9]. The sender crafts the encoded representation of the nonce point R such that it corresponds to a particular (bit) pattern like, for example, the last byte of the point being equal to the subliminal information. Since computing discrete logarithms in finite fields is unfeasible, the sender cannot choose a nonce value, for which the resulting point R has the desired properties. Randomly picking values for the nonce, however, the sender will eventually find a value so that the nonce point shows the desired pattern. In this way, the sender has to test 2^{B_s} distinct nonce values on average, where B_s denotes the desired bandwidth of the subliminal channel in bits. Since the number of computations increases exponentially with the subliminal bandwidth B_s, only a small part of the signature can be used as subliminal channel, which explains the classification as narrowband channel.

[9] Simmons, "Subliminal Communication Is Easy Using the DSA".

This narrowband subliminal channel represents a general approach that can be used for many signature schemes that either explicitly consume randomness for signature generation or implicitly allow many valid signatures for the same message. An advantage of the narrowband channel compared to the broadband channel is that the subliminal receiver does neither have to know the signing key nor the signed message. This narrowband channel is exploitable even if the sign-then-encrypt paradigm is used, under the mild assumption that the subliminal receiver can locate the ciphertext of the signature in the encrypted data. In this case the subliminal sender additionally needs to encrypt the nonce point R (together with the message) after selecting a nonce r, and then checks whether the desired (bit) pattern occurs in the encrypted nonce point (and repeat otherwise).

13.2 Subliminal Communication Scenarios

In two subliminal communication scenarios (multicast clock synchronization and sensor data collection in Smart Grids) the authenticity of the data is of higher concern than their confidentiality so that data may be sent unencrypted (or the encypt-then-sign paradigm could be used). In the blockchain scenario (Section 13.6), confidentiality is even completely undesired as the distributed ledger's content is supposed to be public in the first place. In addition, we show a fourth scenario: the use of subliminal channels in TLS, where the signature is part of the encrypted information. In all four scenarios, subliminal channels can pose a severe threat to information security.

13.2.1 Clock Synchronization

An unrestricted-time high-speed signing scheme based on EdDSA seems particularly well-suited for securing multicast clock synchronization as it provides suitable performance for signing and verification and achieves good security properties (as highlighted in Subsection 8.3.3 and Chapters 9 and 10). Nevertheless, the possibility of subliminal information embedded in the signatures has to be taken into account in security-critical environments. A signature scheme may be an attractive candidate for carrying subliminal information because it can yield a large bandwidth due to the large number of packets (specifically in the context of clock synchronization). Furthermore, clock synchronization protocols are widely deployed throughout the Internet, which leads to a broad infrastructure usable for leaking information through subliminal channels.

In broadcast mode, clock synchronization messages are broadcast in regular intervals across a network. As these broadcasts occur in regular intervals, the amount of data that can be transferred using the subliminal channel is large (when observing a large-enough time span). Exemplary use cases for such a subliminal channel are the clandestine leakage of information through a company's network or the operation of a botnet where the signatures of NTP messages are exploited to transmit C&C messages to bots implemented in NTP clients. As example, Fig. 13.1 depicts the operation of a botnet. If the adversary has managed to install malware on many network nodes and also has infected the NTP server, the subliminal channel can be used for transmitting C&C messages to bots. Approaches to detect the botnet by discovering the C&C communication are then prone to fail.

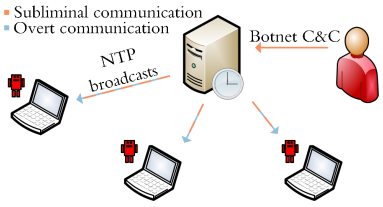

Figure 13.1: Subliminal communication scenario: Botnet C&C using NTP broadcasts.

Adapted from [VI]. © ACM 2018.

13.2.2 Sensor Data Collection in Smart Grids

Unrestricted-time high-speed signing can be used to protect data sent from sensors to data collectors in Smart Grid monitoring. Depending on the amount of sensor data, many signatures need to be sent in short time intervals. Again, data origin authentication is of more concern than confidentiality since modified sensor data may lead to wrong control decisions. Also, for such scenarios we propose the use of unrestricted-time high-speed signing. As an example, we show the use of EdDSA for transmitting phasor measurement data in a Smart Grid environment, sending 60 to 120 packets per second and therefore providing a large bandwidth for subliminal information.

In such a setting there are several reasons for why a subliminal channel has a significant impact on information security. The communicating partners often store sensible data like maintenance schedules, configuration parameters, or even secret key material on the device or accessible by the device. Among others, these data can be used for preparing an attack on critical infrastructure, i.e., the power grid. Furthermore, some real-time applications require data to be transmitted at a high frequency. Signing each of these packets individually, a vast subliminal bandwidth results for data exfiltration. Finally, due to the widespread deployment of sensor and or other Smart Grid components, an adversary finds a large infrastructure for mounting attacks. The homogeneous hardware and configuration of many of these devices allows malware to spread more easily.

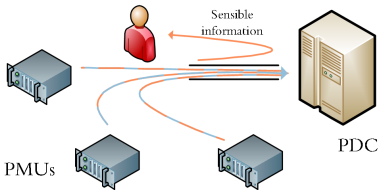

Figure 13.2: Subliminal communication scenario: Information leakage through phasor measurements.

Adapted from [VI]. © ACM 2018.

An example of Smart Grid applications where signatures can lead to a vast bandwidth for data exfiltration is the transmission of measurements by PMUs. Confidentiality of phasor measurements is of lesser importance and encryption may therefore be omitted (or the encrypt-then-sign paradigm be used). The use of high-performance signature schemes for signing measurement transmissions seems natural.

13.2.3 TLS Key Exchange

TLS is the prevalent protocol for securing communications on today's Internet. It is widely deployed and therefore an attractive carrier for information hiding use cases. The subliminal bandwidth is significantly lower than in the two other scenarios described above. Nevertheless, we consider this subliminal channel to be important because of the tremendously wide deployment of TLS. EdDSA is one of the new signature schemes available in TLS 1.3[10] and is proposed to be used also with TLS version 1.2[11,12].

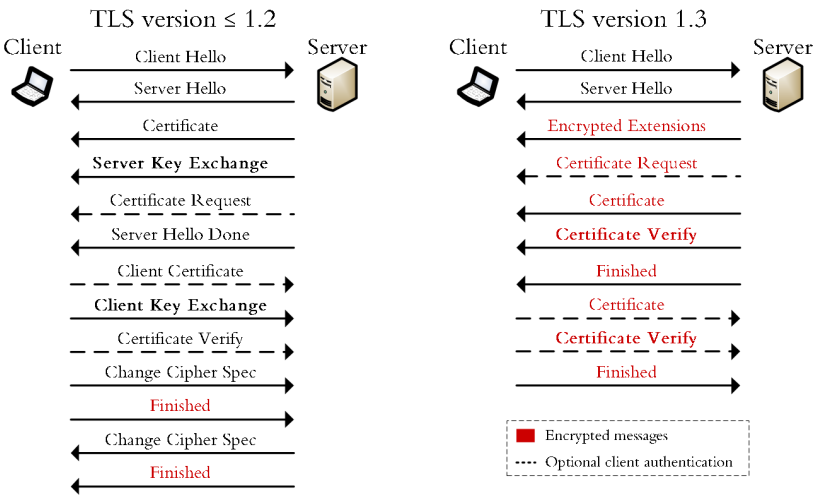

Figure 13.3: Handshake in TLS version 1.3 and before.

Adapted from [VI]. © ACM 2018.

During a TLS handshake, signatures are used to prove the identity of the server and (optionally) of the client. Fig. 13.3 shows the messages exchanged during a handshake. In TLS 1.2 and before, the signature is sent together with the server's Diffie-Hellman parameters in the *Server Key Exchange* message. Besides these parameters the signed data only contain the random values from the client's and the server's *Hello* messages. At that point the exchange is unencrypted and the signed message (which is needed to recover subliminal information in the signature) is known to anyone eavesdropping on the connection. Furthermore, the inclusion of random data makes the detection of the subliminal channel harder. When the client is authenticated using a certificate, the corresponding signature is transmitted in the *Certificate Verify* message. Subliminal channels then exist in both directions, instead of only the subliminal channel from the server to the client.

In contrast to earlier TLS versions, the use of ephemeral Diffie-Hellman is enforced in TLS 1.3 with the associated parameters being exchanged already in the client's and the server's *Hello* messages. The signatures used for authentication are exchanged in the *Certificate Verify* messages. The most important

[10] Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*.

[11] J. Mattsson and D. Migault. *ECDHE_PSK with AES-GCM and AES-CCM Cipher Suites for TLS 1.2 and DTLS 1.2*. RFC 8442 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Sept. 2018. DOI: 10.17487/RFC8442. URL: https://www.rfc-editor.org/rfc/rfc8442.txt.

[12] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246 (Proposed Standard). RFC. Obsoleted by RFC 8446, updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685, 7905, 7919, 8447. Fremont, CA, USA: RFC Editor, Aug. 2008. DOI: 10.17487/RFC5246. URL: https://www.rfc-editor.org/rfc/rfc5246.txt.

difference in TLS 1.3 compared to earlier versions is that the handshake is now encrypted as soon as the shared secret from the key exchange algorithm is available. Therefore, all messages following the *Server Hello* remain unreadable without knowledge of the encryption key. For this reason, the subliminal information cannot be recovered either (in TLS 1.3).

13.2.4 Further Subliminal Communication Scenarios

In addition to the subliminal communication scenarios investigated in this chapter, further scenarios for the use of the subliminal channel in EdDSA are likely and should be taken into account. Subliminal channels could be used to encode additional information in passports[13] or health insurance cards. The issuer could include subliminal information that provides additional information without the owner's knowledge. Also, both DNSCrypt[14] and DNSSEC[15] support EdDSA, which is why these protocols are susceptible to subliminal channels. As DNS lookups occur frequently, a large subliminal bandwidth is possible. Furthermore, upcoming efforts to secure Internet Inter-AS routing in the Border Gateway Protocol (BGP) use nested signatures for path validation. A subliminal channel provides the possibility for clandestine information exchange between BGP routers.

[13] Bohli, Vasco, and Steinwandt, "A subliminal-free variant of ECDSA".

[14] https://dnscrypt.org/

[15] O. Sury and R. Edmonds. *Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC*. RFC 8080 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Feb. 2017. DOI: 10.17487/RFC8080. URL: https://www.rfc-editor.org/rfc/rfc8080.txt.

13.3 Practical Experiments with Subliminal Communication

We perform experiments for three subliminal communication scenarios described in the previous section to get an impression of the difficulty of using the subliminal channel in practice and to analyze the subliminal bandwidth with which data can be leaked.

13.3.1 Clock Synchronization: Signed NTP Broadcast

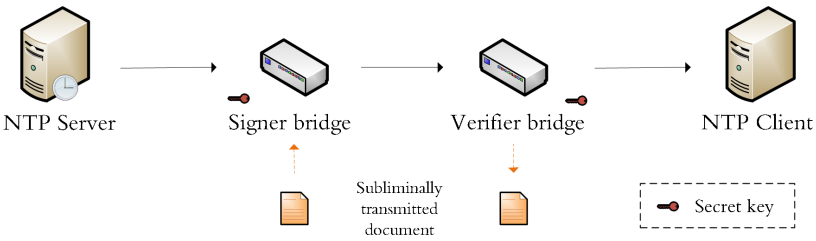

Figure 13.4: Experimental setup for investigating a subliminal channel in signed broadcast NTP messages.
Adapted from [VI]. © ACM 2018.

Fig. 13.4 depicts the experimental setup used for the clock synchronization scenario. The NTP server and the NTP client run unmodified NTP software while the signature generation and verification is performed by the *signer bridge* and *verifier bridge*, respectively, which are located between the NTP server and the NTP client. The subliminal information is embedded by the *signer bridge*, and the subliminal receiver can therefore be anywhere on the broadcast domain.

Server, client, and bridges were running Debian Linux 'Jessie' as operating system and the insertion and removal of signatures was performed with iptables and nfqueue on the network bridges. We used the cryptographic routines from the NaCl[16] library. For recovering the subliminal information (accord-

[16] https://nacl.cr.yp.to/

ing to Eq. 13.1) the library was extended with a routine for performing field subtractions. Apart from this extension, the nonce value was substituted in the signing process to include the subliminal information and was recovered from the signature during signature verification. For sake of simplicity, we transmitted 248 bit per signature instead of 252 bit of subliminal information to avoid dealing with partial bytes. Due to the broadcast interval of 8 seconds, a bandwidth of roughly 3.9 B/s was achieved. 8 seconds is the smallest broadcast interval possible in NTP and, therefore, 3.9 B/s is the largest achievable bandwidth without modifying NTP's source code.

13.3.2 Smart Grid Communication: PMU Sensor Data Transmission

For the second experiment, we used the signer and verifier bridges from the previous setup (Fig. 13.4) to investigate the possibility of subliminal information in signed PMU sensor data transmission. Instead of the NTP server, however, the measurement device employed was a PMU 1133A Power Sentinel by Arbiter Systems, which broadcasts measurement data or sends it to a specific receiver (such as a PDC). PMUs are used to measure the phasors of electric current and voltage at different locations in Smart Grids and send up to 120 packets per second. In this experiment, we used a standard configuration of 10 packets per second and added an EdDSA signature to each packet. We used the manufacturer's proprietary PowerSentinelCSV protocol, which transmits 10 UDP packets of measurement data per second achieving 310 B/s of subliminal bandwidth. In principle the same bandwidth could also be achieved using other protocols for PMU data transmission such as IEEE C37.118, as long as the protocol is available on the device and supports adding signatures.

13.3.3 Network Security Protocols: Key Recovery for TLS 1.2

We examined the EdDSA subliminal channel in TLS from Section 13.2.3 by using a setup consisting of an nginx[17] webserver version 1.13.0 and a simple HTTPS client application, and the BoringSSL[18] TLS library. We chose to use this library because of its support of both Ed25519 and TLS version 1.3. In this case we extended the library by a function that performs field subtractions. Eq. (13.1) was implemented in order to recover the subliminal information, which worked as expected for both TLS 1.2 and TLS 1.3 with a bandwidth of 31 B per handshake[19].

Table 13.1 provides a summary of the measurement results for the three subliminal communication scenarios. Especially with the transmission of sensor data a lot of subliminal data can be transmitted. If client authentication is used in TLS, the subliminal channels can be exploited bidirectionally.

[17] http://nginx.org/

[18] https://boringssl.googlesource.com/boringssl/

[19] In a real-world scenario, session resumption needs to be considered when many subsequent connections occur. In this case, the achievable bandwidth would be lower.

Scenario	Bandwidth
NTP broadcasts	max. 3.9 B/s
PMU measurements	310 B/s
TLS handshake	bidirectionally 31 B

Table 13.1: Measurement results for three subliminal communication scenarios.

Adapted from [VI]. © ACM 2018.

13.4 Preventing Subliminal Communication

Ensuring that a signer does not actively exploit the subliminal channel turns out to be a difficult task. There are some approaches, however, that aim to prevent subliminal communication while retaining compatibility with the signature verification algorithm. In this section, we investigate the suitability of these approaches to prevent EdDSA signatures from being exploited for subliminal communication. Since the subliminal channel in EdDSA proposed in this thesis is employed in the nonce used to generate the signature, the verification process in general would not show any irregularities. Nevertheless, under certain conditions (described below) the existence of a subliminal channel may be suspected though.

13.4.1 Ensuring Subliminal-Free Signatures

If a signature scheme allows only one (valid) signature for a given message, it is possible to ensure that no subliminal information is embedded in a signature. As pointed out by Bohli, Vasco, and Steinwandt, it is also possible to provide subliminal-freeness if the signer can prove to the warden that the signature has been created in a way that permits only one valid signature[20]. Based on this the following countermeasures could be installed to ensure subliminal-freeness.

[20] Bohli, Vasco, and Steinwandt, "A subliminal-free variant of ECDSA".

Pre-published Nonce Points A straightforward solution is to require the signer to generate and publish an ordered list of nonce points to the warden (before the signer knows the subliminal information to be transmitted). During signature generation, the signer must use the nonce points in the same order as they appear in the list. With the nonce being fixed a priori, the signature becomes indeed unique.

The *pre-published nonce points* method has several disadvantages, however. First of all, due to the fixed number of nonce points, the number of messages that can be signed is limited. Secondly, the warden needs to store the list of nonce points, which leads to 32 B storage required for each potential signature. Also, this method introduces a state into signing, which might cause security issues[21]. However, the most important drawback is the fact that also the transmission of the list of nonce points provides a way for embedding subliminal information such that the subliminal channel is just shifted to an earlier time instant. Due to these drawbacks, the use cases for this mitigation method are very limited.

[21] As restoring from backups may result in nonce reuse so that the secret key may be leaked unintentionally.

Warden Interaction Zhang et al. proposed an interactive scheme that turns Schnorr signatures provably subliminal-free, in which the warden actively contributes to signature generation[22]. To prevent a subliminal channel, a total of six messages are exchanged between signer and warden for each signature. The scheme is shown to be secure against existential forgery as long as the computational Diffie-Hellman assumption holds. Furthermore, embedding subliminal information in the signature is shown to be as hard as computing discrete logarithms on behalf of the signer.

[22] Yinghui Zhang et al. "Provably secure and subliminal-free variant of schnorr signature". In: *Information and communication technology-EurAsia conference*. Springer, 2013, pp. 383–391.

Since the EdDSA scheme is based on Schnorr signatures, the *warden interaction* mitigation method is applicable to EdDSA as well. Nevertheless, the major drawbacks of the scheme are the number of messages to be exchanged

between signer and warden for each message and the computational effort required both at the signer and at the warden in order to generate signatures. Furthermore, the warden interaction mitigation requires bi-directional communication between sender and warden, which is not necessarily available in group communication scenarios. These drawbacks conflict substantially with the requirements for data origin authentication in group communication derived in Chapter 6: computational efficiency and low communication overhead.

Zero-knowledge Proofs Bohli, Vasco, and Steinwandt suggested a method for making ECDSA provably subliminal-free, which does not require active participation of the warden[23]. Instead, the nonce used by the signature scheme is generated deterministically from the message and a proof is given to the warden that the value has indeed been derived correctly without providing means for deriving the nonce.

[23] Bohli, Vasco, and Steinwandt, "A subliminal-free variant of ECDSA".

For generating a nonce value, first the hash h of the message m is computed. The nonce is then derived using Naor and Reingold's pseudo random function[24] as

[24] M. Naor and O. Reingold. "Number-theoretic constructions of efficient pseudo-random functions". In: *Symposium on Foundations of Computer Science*. Oct. 1997, pp. 458–467. DOI: 10.1109/SFCS.1997.646134.

$$r(\boldsymbol{h}) = g^{a_{m+1}\prod_{1\leq i\leq m, h_i=1} a_i} \bmod p \bmod L.$$

In this equation, p is a prime number and g is the generator of a cyclic group of prime order q. The vector $\boldsymbol{a} \in \mathbb{Z}_q^{m+1}$ is an additional secret for signature generation.

The signer computes commitments for the additional secret during generation and shows them to the warden. During signing the signer can then compute zero-knowledge proofs that proof (to the warden) that the nonce has in fact been computed correctly. While the signature is guaranteed to be subliminal-free, the proof itself is not and therefore must be stripped by the warden after verification.

This provably subliminal-free signature scheme is formulated for ECDSA only. Since it solves the general problem of showing that a curve point has been generated according to some specific method from the message without disclosing the point's discrete logarithm, however, it can be equally applied to EdDSA. Compared to the warden interaction method, it has the advantage of simplifying the communication pattern between signer and warden. Bohli, Vasco, and Steinwandt proposed to use the scheme for passports where it should be possible for the passport's holder to make sure that the issuing party has not embedded information in the signature. Since one proof takes several megabytes (for a security level of 128 bit), the communication overhead between signer and warden is significant—too large in the context of group communication. The computational overhead is significant as well.

Table 13.2 summarizes the distinct methods for mitigating subliminal communication with respect to their advantages and drawbacks.

13.4.2 Detecting Subliminal Communication

As it is not possible to prevent subliminal channels without introducing significant drawbacks, the question arises if subliminal communication can at least be detected. In this subsection, we highlight situations in which subliminal

Method		Advantages and drawbacks
Pre-published nonce points	+	Simple
	+	Low computational requirements.
	-	Limited number of transmitted messages.
	-	Subliminal channel exists during list computation.
	-	Storage required for warden and sender.
Warden interaction	+	Small bandwidth requirements
	-	Participation of warden required
	-	Several messages need to be exchanged
	-	Need for bidirectional communication.
	-	Subliminal channel exits to and from warden.
Zero-knowledge proofs	+	Simple communication pattern.
	+	Feasible for offline scenarios.
	-	Significant communication overhead.
	-	Significant computational overhead.
	-	Subliminal channel to warden exists.

Table 13.2: Methods to provide subliminal-freeness.

Adapted from [VI]. © ACM 2018.

communication can lead to suspicious communication patterns that may be observed.

Identical Messages Due to deterministic calculation of the nonce, a particular message produces the same signature independent of how often the message is transmitted. If the same message is transmitted multiple times and the nonces have not been derived from the messages but carry (distinct) subliminal information, the subliminal communication can be detected by the fact that the signatures differ although the messages are identical. A warden who monitors the communication can notice that signatures are distinct although they were generated for identical messages under the same key pair. From this observation, the warden may infer that subliminal information has been transferred. With enough effort the subliminal sender can circumvent this detection technique, however. In order to prevent this detection, the subliminal sender needs to check whether an identical message has been sent before. If this is the case, the same nonce should be used as before in order to not raise suspicion. The subliminal receiver can just discard any subliminal information received in duplicated messages. This method increases storage requirements significantly for both subliminal sender, warden, and subliminal receiver.

Small Nonce Values As described in Section 2.6, it is of utmost importance for the security of the signature scheme to sustain unpredictability of nonces. However, when directly encoding (unencrypted) subliminal information into the nonce, it may regularly take small values or even become equal to zero, depending on the subliminal information that is being transmitted. Detection of

such values can, therefore, not only lead to detection of the subliminal channel, but also allow an eavesdropper to recover the signing key. The sender of the subliminal information can mitigate such small nonce values by encrypting the subliminal information such that the nonces will be indistinguishable from random data. In fact, encryption is often performed for subliminal channels to prevent others from being able to read the transferred information anyway.

Repeating Nonce Values As explained in Section 2.6, the signing key can be recovered from signatures for distinct messages when the same nonce has been reused. In order to remain secure, the sender of the subliminal information has to ensure that distinct nonces are used even if the subliminal information is identical. For this purpose, the Output Feedback (OFB) mode of block ciphers can be used to significantly reduce the probability of reoccurring random values[25].

[25] Menezes, Van Oorschot, and Vanstone, *Handbook of applied cryptography*.

Zero-knowledge Proofs In the particular case when the requirements for the warden can be relaxed to prove the existence of a subliminal channel only when examining a random sample of signed messages, zero-knowledge proofs can also be used for a scenario where it is unfeasible to place a warden in a MITM position as long as the signer can be obliged to offer proofs for generated signatures on a protected interface. This is for situations where it suffices to test a random sample of generated signatures for subliminal-freeness. A signature that has been intercepted unnoticeably can then be tested for having been generated correctly. In this case, however, the signer must make sure that the warden already has a valid signature for the message, as the signer would otherwise sign arbitrary messages on behalf of the warden. When the warden wants to check a signature for subliminal-freeness, the warden requests a proof for the signed message. As the nonce is computed deterministically from the message, the signer can recreate the same signature and compute a proof using Naor and Reingold's pseudo random function.

13.5 Subliminal Channels in MQ-based Signature Schemes

13.5.1 Randomness in MQ Signatures

The basic operation principle of MQ signature schemes (Section 2.7) creates the impression that these schemes deterministically map a given message to a signature, leaving no space for subliminal information. Nevertheless, all MQ signature schemes that are considered secure as yet use randomness throughout the signing process. Methods for introducing randomness in the signing process can be classified into two groups: (1) include random data in the signature to achieve security (Unbalanced Oil and Vinegar (UOV) trapdoor, the minus modification, and the vinegar variables modification), and (2) reduce the probability of the central mapping to be invertible for a message such that randomness has to be included to be able to find a signature for a particular message. The HFE trapdoor, the fixing modification, the internal perturbation modification, and the plus modification cause the trapdoor to not be surjective, which means that there would exist messages for which there is no signature. In order to guarantee a signature to exist for every message with sufficient probability, randomness has to be included during signature generation

by using either the minus modification or the vinegar variables modification. Surprisingly, modifications that lead to the loss of surjectivity can not only create a subliminal channel but may also reduce the subliminal bandwidth. Subliminal bandwidth may be reduced because the probability of finding a signature (with a particular choice of random data) is lowered such that the signer needs to vary more variables in order to find a signature with sufficient probability. Still, use of these modifications for the sole purpose of reducing subliminal bandwidth is not justified as the modifications significantly reduce signing speed.

13.5.2 Subliminal Channels in MQ-based Signature Schemes

To get an understanding of how much data can be exfiltrated using subliminal channels in MQ signature schemes we analyze algorithms for which an implementation exists or at least a practical set of parameters has been proposed. It needs to be stressed that we aim at analyzing techniques that are used for constructing MQ signature schemes rather than concrete algorithms. Hence, even though many of the schemes described below have been broken, the results have a certain relevance for signature schemes that are to be developed in the future and are likely to be constructed similarly. Table 13.3 shows the results. For a detailed discussion on subliminal channels in MQ-based signature schemes see[26].

[26] Alexander Hartl, Robert Annessi, and Tanja Zseby. "Subliminal Channels in High-Speed Signatures". In: *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)* 9.1 (Mar. 2018), pp. 30–53.

Scheme	Trapdoor	Broken	Signature length	Subliminal bandwidth	
QUARTZ	HFEv-	no	128 bit	∼ 12 bit	(9%)
Gui-127	HFEv-	no	163 bit	∼ 24 bit	(15%)
SFlash	C^*-	yes	469 bit	77 bit	(16%)
PFlash(GF16,94,30,1)	pC^*-	no	372 bit	∼ 108 bit	(29%)
MQQ-SIG	MQQ-	yes	256 bit	128 bit	(50%)
Rainbow	UOV-, STS	yes	264 bit	∼ 46 bit	(17%)

Table 13.3: Subliminal bandwidths of MQ signature schemes.

Adapted from [VII]. © JoWUA 2018.

13.5.3 Discussion

All methods for constructing MQ schemes that are considered secure as of today provide significant bandwidth for subliminal communication. It is noteworthy that subliminal channels in MQ-based signature schemes may have a particularly attractive property: in contrast to broadband subliminal channels of DSA-like signature schemes, the possibility to decode the subliminal information does not directly coincide with the possibility of signing messages. Using a broadband subliminal channel in ECDSA or EdDSA, forces the subliminal sender to share the signing key with the subliminal receiver, who could then forge signatures on behalf of the sender. For subliminal channels in MQ-based signature schemes, however, the subliminal sender can choose not to use the entire subliminal bandwidth in order to keep a level of security against forgery of signature by the subliminal receiver.

If the entire subliminal bandwidth is used, the subliminal receiver can forge signatures. By using just part of the subliminal bandwidth, however, a trade-

off can be achieved between subliminal bandwidth and security against attacks performed by the subliminal receiver. Furthermore, by passing on different parts of the set of hidden equations, it is possible to transmit different subliminal information to multiple receivers, who are unable to decode the information that is not intended for them. We note that mitigation strategies that use zero-knowledge proofs as described in Section 13.4 are significantly more difficult to construct for MQ signatures schemes because many zero-knowledge proofs rely fundamentally on the hardness assumption of discrete logarithms. For post-quantum cryptography (one of the use cases for MQ signature schemes) this assumption can no longer be assumed to hold.

13.6 *ChainChannels: Subliminal Channels in Blockchains*

Attribution to paper coauthors

The work on ChainChannels (described in the joint paper [V] and Section 13.6) was conducted in collaboration with Davor Frkat. The practical implementation was conducted primarily by Davor Frkat.

In this section, we present *ChainChannels*, a novel way of secretly multicasting information over public blockchains that can be used for the distribution of C&C commands to associated bots for example. For this purpose, subliminal information is embedded in the signatures used to secure blockchain transactions. Since signatures are essential for the operation of blockchains, they provide a transmission method that can be exploited by botnet operators. We show how signatures in blockchain transactions can be used to transmit subliminal information and also show how the keying material (needed to extract the subliminal information) can be distributed securely to bots such that they do not need to store the secret key in advance and a take-over by an adversary is prevented that acquired information from a compromised bot. In ChainChannels, an adversary can only follow the communication with a compromised bot but cannot take control over the botnet.

The two main digital signature schemes used for blockchain applications are ECDSA and EdDSA. For both it has been shown that they contain broadband subliminal channels, but also other signature schemes that could be used in blockchain applications may allow the injection of subliminal information. Since the secret key is used to sign transactions in the blockchain, sharing the secret key enables receivers of the subliminal information (i.e., the bots) to forge transactions on behalf of the sender. To prevent such forging of transactions, we will introduce a method to distribute the secret key without a receiver being able to forge transactions when they are still accepted as valid by bots.

13.6.1 *Assumptions*

In the following we often refer to the Bitcoin blockchain as an example. Nevertheless, ChainChannels is equally applicable to any other blockchain as long as its signature scheme allows the injection of subliminal information. We make the following assumptions, which we find to be valid and to not reduce the generality of the results:

1. We consider the initial botnet establishment as out of scope, i.e., scanning for vulnerable hosts, compromising hosts, and propagation of botnet software to compromised hosts is not described here and can be done by any method nowadays used. Also, any potential upstream communication from

the bots to the botmaster is considered out of scope.

2. We assume that bots are connecting directly to the blockchain and extract the signatures from the transactions. Alternatively, the bots can use blockchain explorer websites (or their respective APIs), which can be implemented to reduce communication overhead or as a fallback option when bots do not have access to the blockchain due to firewall restrictions for example.

3. Bots are provided with an initial address, on which they check for new commands. This initial address can be provided with the installation of the botnet software. In ChainChannels, knowledge about this initial address does not help an adversary to find the C&C or information about the botnet structure (after a bot was taken over for example).

The two building blocks required to establish the subliminal channel for a C&C infrastructure are the following: (1) a signature scheme such as ECDSA or EdDSA that allows injecting subliminal information, and (2) a method to secretly exchange information required to use the subliminal channel, i.e. in our case this is the distribution of the secret key to the bots. Both building blocks are described in detail in Sections 13.6.2 and 13.6.3.

13.6.2 Subliminal Channels in Blockchain Signatures

As an example we use the public blockchain for the Bitcoin cryptocurrency. Like many other blockchains, the Bitcoin blockchain uses the ECDSA signature scheme with curve secp256k1[27]. Since there are many variations and developments stemming from Bitcoin, the introduced subliminal channel is not limited to the original Bitcoin blockchain and independent of other design decisions. For secp256k1 the initial parameters, the generator point G, and order of the curve n are publicly known. An ECDSA signature consists of the tuple (r, s), which is calculated for each message as follows[28]:

1. A cryptographically secure nonce k is generated with $1 \leq k \leq n - 1$. Measures have to be taken so that k can not be guessed by an adversary, which are left to the implementation (see Section 2.6).

2. A new point on the curve is computed by $(x_1, y_1) = k \cdot G$. The first coordinate (x_1) is used as first part of the signature, i.e., $r = x_1$.

3. The message m to be signed consists of a raw preliminary transaction[29]. The message is hashed twice with SHA-256, referred to as hashed message z.

4. The second part of the signature (s) is then calculated from the secret key d, the hashed message z, the first part of the signature r, and the multiplicative inverse of the nonce k as follows:

$$s = k^{-1} \cdot (z + r \cdot d) \mod n \tag{13.2}$$

In order to insert subliminal information in the signature, the nonce k, which is supposed to be random, is substituted with the subliminal message. The signature carrying the subliminal message is then generated as usual, just

[27] Daniel R. L. Brown. "SEC 2: Recommended Elliptic Curve Domain Parameters (Version 2.0)". In: *Standards for Efficient Cryptography Group (SECG)* (Jan. 2010).

[28] Johnson, Menezes, and Vanstone, "The elliptic curve digital signature algorithm (ECDSA)".

[29] *Bitcoins the Hard Way: Using the Raw Bitcoin Protocol [Online].* http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html [Accessed: 18-Sep-2018].

that the nonce k now contains subliminal information instead of a random number. In order to extract the subliminal information from the signature the receiver needs to know the secret key d. With knowledge of the secret key, the subliminal information k_{msg} then can be extracted by the receiver as follows:

$$k_{msg} = s^{-1} \cdot (z + r \cdot d) \mod n \tag{13.3}$$

While the signature is valid and independent of the particular nonce, it must be avoided to use repeating values of the nonces under the same secret key. Repeating values of the nonces may occur if identical (unencrypted) C&C messages are transmitted. If the same nonce is used multiple times, the secret key is leaked unintentionally[30].

In Section 13.6.3, we show how the sender can distribute the secret key such that there is no need to pre-configure the bot with the secret key. Storing the secret key into the bot would imperil the whole botnet. Instead, we implement mechanisms to provide the entire secret key only after it is safe to assume that the embedded subliminal message can not be altered anymore.

[30] Johnson, Menezes, and Vanstone, "The elliptic curve digital signature algorithm (ECDSA)".

13.6.3 *Distribution of the Secret Key*

The propagation of a subliminal message is depicted in Fig. 13.5. Subliminal messages are pushed to the blockchain by the botmaster and the bot knows from which address to expect the subliminal botnet commands. Nevertheless, the bots require knowledge of the secret key d used to sign messages in order to extract the subliminal message. In this subsection, we present different methods how the secret key can be distributed to the bots.

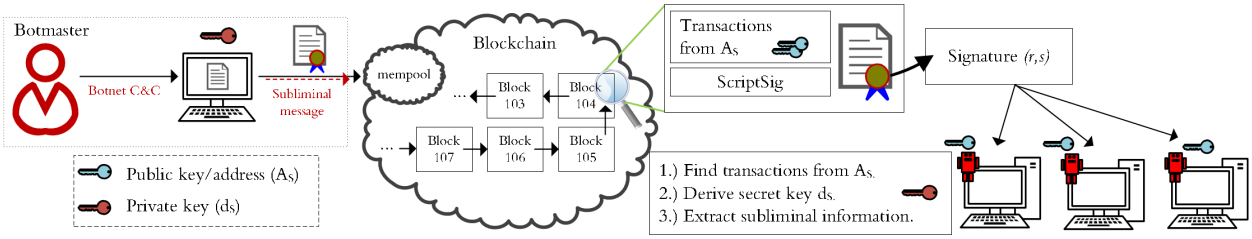

Figure 13.5: ChainChannels: Subliminal message propagation The transaction is signed by the botmaster with a determined nonce as subliminal message. The transaction can be seen in the mempool or as part of a mined block and is found by its source address and processed by bots. In the final transaction, the botmaster uses a nonce that is pre-configured in the bots. The bots can therefore derive the secret key and extract the subliminal information encoded in the previous transactions.

Adapted from [VII]. © IEEE 2018.

Pre-Configuration of the Secret Key A naive method is to simply pre-configure the secret key in the bots (e.g., as part of the botnet code). Knowing the secret key is sufficient to deduce the public key and with this the sender address. Nevertheless, with this method an adversary (to the bot owner) just needs to compromise a single bot to gain knowledge of the secret key and therefore can take over the botnet since messages can be crafted that are indistinguishable from messages by the actual botmaster.

Key Leakage Based on Nonce Reuse A better method to distribute the secret key to the bots is to send the key **after** the botnet is created. In this way, the key needs to be revealed only shortly before the botnet commands are executed. To this end, the bots first collect all messages received from the botmaster without being able to extract the subliminal information. As soon as the bots receive the secret key, they extract the subliminal messages and then trigger the

execution of commands. In this way, the secret key remains unknown to the bots and undetectable for an adversary until botnet commands are triggered.

$$d = \frac{m_1 \cdot s_2 - m_2 \cdot s_1}{r \cdot (s_1 - s_2)} \mod n \tag{13.4}$$

The second method to distribute the secret key is to reuse nonces deliberately. Since the first part of an ECDSA signature (r) depends only on the nonce and the generator point G, nonce reuse requires two signatures (r, s_1) and (r, s_2) for two distinct messages m_1 and m_2 but with identical values of r. After nonces have been reused, the secret key can be calculated according to Eq. (13.4). The bots are checking all transactions from the sender address and analyze the nonce points. When transactions with reused nonces are observed, the bots are able to calculate the secret key from the two signatures, can extract the subliminal messages, and thereafter trigger the execution of botnet commands.

Bad random number generators may also generate identical nonce values accidentally, which lead to the same r values in distinct signatures. Anyone who finds such duplicate nonces can derive the secret key as well and can use it for instance to steal Bitcoins. Therefore, the detection of nonce reuse is a well-established method and tools exist to search for identical r values in signatures. For this reason, one drawback of the key leakage method based on nonce reuse is that other blockchain users may notice it as well by parsing the blockchain and may therefore derive the leaked secret key. Subsequently, the messages transmitted to the bots would be publicly accessible. For this reason, it is beneficial to distribute the secret key in a way that only the bots can decode it (shown next).

Key Leakage Based on Secret Sharing To solve the problem of exposing the C&C messages to the public, a secret sharing scheme such as Shamir's[31] or Brickell's[32] can be used. For the problem at hand any secret sharing scheme can be employed without affecting the basic principle of operation. For a simple secret sharing scheme the secret key d is split into two parts X_1 (Eq. 13.5b) and X_2 (Eq. 13.5c). For this purpose, a cryptographically secure random number R is needed. A large prime n is used for the modulo operation, which defines the upper limit of d and R (see Eq. 13.5a). Since the size of the key to be split is 32 byte, the (prime) order n of the secp256k1 curve[33] can be used as large prime n. The secret key d can be reconstructed efficiently when both parts, X_1 and X_2, are known (Eq. (13.5d)).

[31] Shamir, "How to Share a Secret".

[32] Ernest F. Brickell. "Some Ideal Secret Sharing Schemes". In: *Advances in Cryptology — EUROCRYPT '89*. Lecture Notes in Computer Science. Springer, Apr. 1989, pp. 468–475. ISBN: 978-3-540-46885-1.

[33] Brown., "SEC 2: Recommended Elliptic Curve Domain Parameters (Version 2.0)".

$$0 < d < n \text{ and } 0 < R < n \tag{13.5a}$$

$$X_1 = (d + R) \mod n \tag{13.5b}$$

$$X_2 = (d + 2R) \mod n \tag{13.5c}$$

$$d = (2 \cdot X_1 - X_2) \mod n \tag{13.5d}$$

We assume that all bots know the initial address A_L from which the key will be leaked and also know the value X_1 initially, which can be distributed to the bots in the same way as the initial address. The difference to the key leakage based on nonce reuse is that bots know only the common X_1 and the information leaked (X_2) is useless without the common secret X_1. In

this way, a part of the secret can simply be pre-configured in the botnet code without the danger that an adversary gets knowledge of the entire secret key as soon as one bot gets compromised. In this way, the first part of the secret (X_1) is required to calculate the entire secret key while only the second part of the secret (X_2) is leaked through nonce reuse.

One important detail with all secret sharing schemes is that the address from which X_2 is leaked has to be different from the address from which the subliminal messages are sent. The reason for this is that with the leaked information the secret key of the leaking address can be derived. If there are two equal r values observed from address A_L anyone can derive the secret key d_L for A_L, which means that an adversary could wait for the key leakage, derive the secret key, and then inject subliminal messages to take over the botnet. If the subliminal messages are sent from a different address (e.g., A_S), however, then X_1 is required together with the X_2 to calculate the secret key d_S of A_S (Eq. 13.5d), and only bots that had the X_1 pre-configured can decode the subliminal messages. The processing steps for key leakage based on secret sharing are shown in Fig. 13.6.

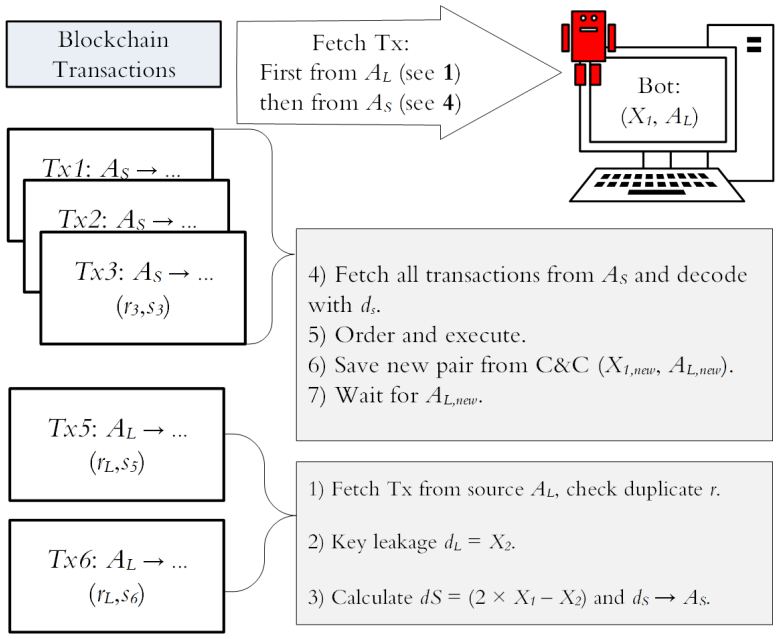

Figure 13.6: ChainChannels: Key leakage based on secret sharing. Transaction and processing steps at a bot using key leakage based on secret sharing. Here only the transactions from the address that injects the subliminal information A_S (Tx1-3) and from the address (A_L) that leaks the key are shown. The transactions from A_S are inserted in the blockchain before the key is leaked. The transaction receivers are arbitrary and can vary. The bot first fetches transactions from the leaking address A_L, and check for duplicate r that leak the secret key (d_L) of A_L. It uses the leaked X_2 (which corresponds to d_L) to calculate the secret key d_S. From d_S the bot can derive the address of the subliminal sender A_S. With d_S the bot can then decode the subliminal messages from transactions from A_S.

Adapted from [VII]. © IEEE 2018.

Concealed Key Leakage With a Pre-Shared Nonce Until now, we have secured the subliminal messages from being directly exposed to the public by not pre-configuring the (entire) secret key in the bots. One problem remains though: the existence of nonce reuse alone may raise suspicion, especially given the fact that blockchains may be monitored for nonce reuses, mainly with the aim to steal coins associated with a leaked key[34]. While not dangerous to the operation of the ChainChannel's C&C infrastructure, nonce reuse may draw attention to the botnet, which is undesired. To stay covert and to not produce suspicious transactions, we propose to use an advanced method to avoid reusing nonces.

For this purpose, we intentionally violate the assumption that the nonce must be unpredictable. We suggest for ChainChannels to pre-configure a

[34] Joppe W. Bos et al. "Elliptic Curve Cryptography in Practice". In: *Financial Cryptography and Data Security*. Lecture Notes in Computer Science. Springer, Mar. 2014, pp. 157–175. ISBN: 978-3-662-45472-5.

random secret nonce k_{leak} at the bots (in addition to the initial address). The pre-configured nonce is known only to the bots and the botmaster and remains unpredictable to everyone else. In addition, we do not need to use different addresses for key leakage in this case but can do the key leakage from the same address from which the subliminal messages are sent (i.e., A_S).

The botmaster uses transactions from address A_S to send commands and to leak the key. After all commands have been inserted and confirmed in the blockchain, the botmaster uses the pre-shared k_{leak} to generate the final signature (r_{leak}, s_{leak}) and, in this way, leaks the secret key needed to decode the commands. The bot knows the initial address A_S and the pre-configured nonce k_{leak}. It checks all transactions from A_S and extracts the signatures. For each signature (r, s) the bot checks whether the pre-shared nonce k_{leak} has been used by verifying that $r == r_{leak}$ (and r_{leak} can be precalculated as $k_{leak} \cdot G$). If the key leakage transaction has been detected, the secret key for A_S can be derived by Eq. 13.6. The processing steps for the concealed key leakage are shown in Fig. 13.7.

$$d = r^{-1} \cdot (s \cdot k_{leak} - z) \mod n \tag{13.6}$$

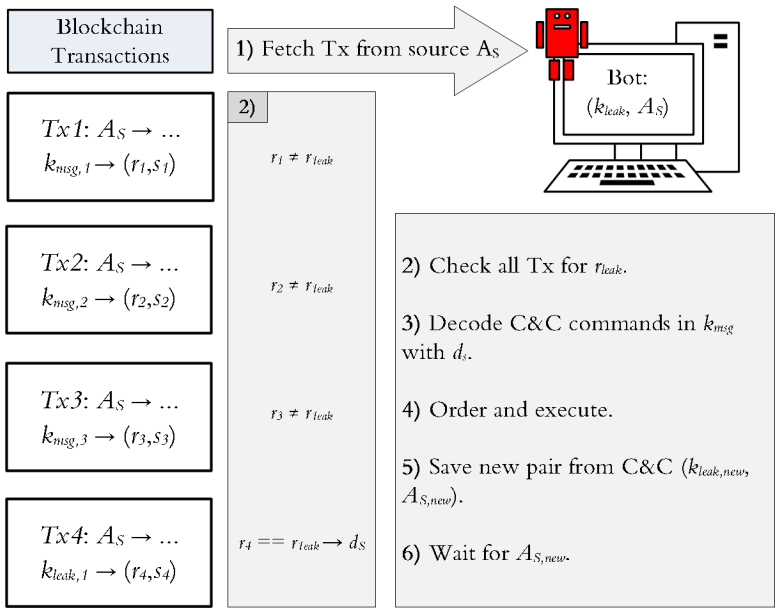

Figure 13.7: ChainChannels: Concealed key leakage. Transaction and processing steps at a bot using concealed key leakage. First the subliminal messages are injected by address A_S and eventually a signature is generated with k_{leak} to leak the secret key d_S of A_S. The bot continuously fetches transactions from A_S and checks whether the signature was generated with k_{leak} in order to derive the secret key d_S of A_S. The key d_S can then be used to decode the commands in the previous transactions.

Adapted from [VII]. © IEEE 2018.

To an observer the signatures of the ChainChannels C&C infrastructure are indistinguishable from signatures of regular transactions as they do not trigger any key leakage detection. The bots, however, who know the address and the pre-shared nonce can derive the secret key and therefore extract the subliminal information included in the signatures of previous transactions. In order to keep ChainChannels functional for multiple botnet commands, the subliminal information in the messages from the initial address must entail a new address to listen for transactions as well as a new pre-shared nonce. In this way, ChainChannels can be used for the lifetime of the blockchain.

Cross-Blockchain Usage Since ChainChannels does not depend on features of a particular blockchain, it can even be distributed over multiple blockchains.

The blockchain could be changed from one botnet command to the other in order to make use of features of a specific chain such as low transaction fees or low block time (and therefore latency) or even to just further obfuscate the plain existence of the botnet.

13.6.4 *Proof of Concept*

In order to demonstrate the applicability of ChainChannels we crafted transactions that we used to inject a subliminal message into the Bitcoin blockchain. We used the open source Bitcoin client *Electrum*[35] and modified its code to embed our subliminal message. As this is meant to be a proof of concept, we implemented the show case of embedding a subliminal message with concealed key leakage in the Bitcoin blockchain. A full-fledged implementation, however, should also prompt for commands, handle key usage, encoding, and address usage to fit the C&C infrastructure as well as handle various blockchains.

[35] Available under https://electrum.org.

For our proof of concept the values listed in Table 13.4 were used, with A_S being the address that leads to the transactions with the subliminal message and concealed key leakage and k_{leak} denotes the pre-shared nonce for address A_S. Other values such as the signature pair (r_i, s_i) or the transaction hash z_i can be derived from the transactions. All transactions before the final transaction include subliminal information that is stored (and extracted later). In the final transaction from A_S the concealed key leakage takes place. For this purpose, the secret key d_S can be derived and the subliminal message encoded in the nonces of the previous transactions can be extracted (Eq. 13.6). To make the proof of concept comprehensible, we encoded the subliminal message as 8-bit ASCII plaintext. With the information provided in Table 13.4 the reader can simply fetch the transactions, calculate the leaked secret key with the known nonce, and decode our subliminal message from the public Bitcoin blockchain.

Description	Symbol	Value
Source Address (message & leakage)	A_S	1Ahg5AkeNJHorpfvzUmGRqNZgGtC9BGzdQ
Pre-shared secret nonce for leakage	k_{leak}	0x51b094cca21e39f76f50486841a315284669ff0f2b3481b9720e8d18443a2ee7
Secret key subl. message (calculated)	d_S	0x3e77c102528282cc62639755438b1f541d48c9a87130ead96772d939b9a4ec95
Order of curve (secp256k1	n	0xfffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141
TxID subl. message (block 508748)	$TxID_1$	0xbfdd501624f574662f67a57aa0b1db48bfa82f8a2c949ed8978970c49a76cf92
TxID key leakage (block 527646)	$TxID_2$	0x4194598850970792127 8fbff9254d468e634062b82fa5459c3e8bf12909a1955

Table 13.4: Values for ChainChannel's proof of concept.
Adapted from [VII]. © IEEE 2018.

13.6.5 *Discussion*

An important feature of ChainChannels is that it solely relies on digital signatures. It does not need fields of a particular blockchain implementations such as the OP_RETURN field in Bitcoin and is therefore applicable to arbitrary blockchains. Not only the content of the communication is hidden but also the that fact that communication between the botmaster and the bots took place at all. Furthermore, the number of active bots cannot be revealed by monitoring the C&C infrastructure since it is not recorded who queries which part of the blockchain. Nevertheless, there are a few factors that influence the usability of ChainChannels, which we will discuss next.

Transaction Fees In blockchains for cryptocurrencies transactions cost fees, which depend on the size of the transaction (and on the particular blockchain). For this reason, a botmaster has costs for operating the botnet (i.e., executing commands). Nevertheless, the development and maintenance of any conventional or neoteric C&C infrastructure produce costs in effort and money as well[36]. Additionally, botnets are exposed to the risk of take-down, which would render the initial effort useless. ChainChannels scales perfectly with the number of bots and has excellent availability due to the flexibility provided by employing multiple blockchains and the blockchains themselves being decentralized. Furthermore, to take down the botnet the corresponding blockchains need to be taken down in the first place, which significantly reduces the botnet's take down risk. Also the applicability to different blockchains provides the botmaster the choice to use blockchains with low fees. We therefore think that transaction costs are only a minor drawback.

[36] Vormayr, Zseby, and Fabini, "Botnet Communication Patterns".

Bot Takeover An adversary who is able to take over a single bot is able to retrieve the current (k_{leak_i}, A_i) pair. With knowledge of the current pair, the adversary is able to get the leaked secret key and can listen to the C&C communication (from that on, but not previous commands). Also, one could argue that an adversary in possession of the leaked key may be able to forge own messages and send own commands to the bots. Such forged messages could for instance contain fake future addresses where no new messages can arrive in order to render the botnet useless. In order to prevent this, we propose to first inject the commands in the blockchain and to leak the key only after the corresponding blocks have been confirmed. In this way, the key leakage is solely used as a trigger after all other transactions (i.e., the botnet commands) are pushed. An adversary can therefore inject own commands into the blockchain only after the key has been revealed to the bots, which means that the bots are already listening for transaction with the new source address and ignore commands from the old address to which the adversary has access.

Transaction Malleability Parts of the ECDSA signatures in Bitcoin and similar blockchains can be changed without voiding their validity, as presented for EdDSA in Section 13.1. Furthermore, numerous attacks occurred where this was abused. Transactions were intercepted, modified, and transmitted by the attacker to pass the same transaction with another transaction ID[37]. Bitcoin therefore enforces low values of s to combat signature malleability. If a value s is larger than half the curve order n, the value $s_{new} = n - s$ is used instead[38]. The signature is still valid, but the subliminal channel cannot be calculated as described in Eq. 13.3. When decoding a subliminal message, the s value has to be checked for the mentioned condition, and changed as needed to ensure reliable decoding of messages.

[37] Christian Decker and Roger Wattenhofer. "Bitcoin Transaction Malleability and MtGox". In: *Computer Security - ESORICS*. Lecture Notes in Computer Science. Springer, Sept. 2014, pp. 313–326. ISBN: 978-3-319-11212-1.

[38] Pieter Wuille. *BIP 0062: Dealing with Malleability - (2014) [Online]*. https://github.com/bitcoin/bips/blob/master/bip-0062.mediawiki [Accessed: 18-Sep-2018].

Latencies Ali et al. measured that about 90% of the bots respond in 10 seconds[39]. Since the bots can see the transactions while they are in the mempool awaiting confirmation, they do not necessarily have to wait until they are included in a block. With ChainChannels, some preparation time has to be accounted for since the transactions containing the subliminal C&C messages have to be confirmed first before leaking the key. An adversary could try to

[39] Ali et al., "ZombieCoin 2.0".

attack the botnet if the key is leaked too early by offering higher transaction fees than the botmaster. Nevertheless, after the key leakage is performed, the bots can be triggered in the same time span.

Selection of Suitable Blockchains Suitable blockchains for ChainChannels can be determined by the following factors: current transaction fees, number of transactions per time, confirmation speed (i.e., block time), signature scheme, and anonymity.

Nonce Randomness In our proof-of-concept we injected a subliminal message as (ASCII encoded) plaintext in the nonce of the signatures. Such encoding reduces the overall range of possible nonce values and produces bias in the distribution of nonce values such that they can no longer be considered random. Since using a non-random nonce weakens the security of the signature, we suggest encrypting the subliminal message before it is injected in the signature. In this way, nonces remain cryptographically secure and biased patterns are avoided. As symmetric encryption key the secret key can be used that is known to the bots anyway after the key leakage.

13.6.6 Detection and Mitigation

The most effective way to take down ChainChannels is to prohibit blockchain traffic in general or take the particular blockchain down, which requires immense effort and is therefore highly unlikely. Prohibiting blockchain traffic might work for networks with strict firewall rules, such as corporate networks. Communication could still be conducted over public blockchain explorer websites, nevertheless.

Signature Specific Mitigation In Section 13.4, we discussed several methods for the detection of subliminal communication. Small and repeating nonce values may pose a threat for the botmaster in terms of detection, which can be caused by sending the same deterministic commands, thus generating small and/or repeating nonce values, and not encrypting the messages. This risk of detection can be avoided by the botmaster by embedding some random padding in the messages or by encrypting the messages before they are injected (as explained before). The other transmitted data, such as $(X_{1,new}, A_{new})$, are supposed to be random anyway so that they do not produce suspicious patterns when embedded in the per-message nonce k.

Employing subliminal-free variants of ECDSA or EdDSA requires a trusted third party (the warden) to not only check the signatures but also participate in the signing process. Having a trusted third party requires a lot of resources and also counters the decentralized and trustless concept of most blockchain applications and is therefore not a viable solution.

Blockchain Specific Mitigation If a subliminal-free signature scheme were to be introduced, this would result in a hard-fork for a blockchain[40]. This means that the update can be introduced by the developers but still must be accepted by 50% plus one node to become standard. Eventually, hard-forks can split up the blockchain, which results in two independent blockchains and can have unwanted economic consequences. Another method to take down a botnet

[40] M. Sato and S. Matsuo. "Long-Term Public Blockchain: Resilience against Compromise of Underlying Cryptography". In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–8.

would be to link the used coins to the botmaster. Methods to deanonymize holders of cryptocurrency are being developed constantly by authorities to battle criminals[41]. Such methods could be utilized to link a botmaster to an exchange, where the authorities could request the identity of a user. Nevertheless, since the destination address in transactions can be arbitrary, the receiver of coins can be a random unsuspecting user, who could eventually send the coins to an exchange, thus driving the attention away from the botmaster.

[41] M. Moeser, R. Boehme, and D. Breuker. "An Inquiry into Money Laundering Tools in the Bitcoin Ecosystem". In: *2013 APWG eCrime Researchers Summit*. Sept. 2013, pp. 1–14.

14
Conclusion

In this thesis, we tackled a fundamental challenge of securing group communication for critical infrastructures: data origin authentication. Although research on data origin authentication for group communication was conducted for more than twenty-five years and many schemes have been published, a thorough understanding of applications' requirements was of crucial importance to the process of designing and selecting suitable data origin authenticating schemes. We established a threat model and derived properties that data origin authentication schemes should provide in order to make them generally applicable to critical infrastructure applications. In this way, we established a comprehensive set of properties in three distinct categories: performance, security, and robustness.

We conducted a theoretical evaluation of data origin authentication schemes for group communication to study to what extent the properties identified before are satisfied. Our evaluation showed that each class of schemes comprises a trade-off from a specific point of view. We found that not a single state-of-the-art data origin authentication scheme provides all properties required for making it generally applicable to critical infrastructure applications.

We furthermore analyzed how research on data origin authentication schemes for group communication evolved over the last twenty-five years. We found that the state-of-the-art classification does not cover research developments in recent years sufficiently and suggest an improved classification. We identified three basic approaches to data origin authentication: extending symmetric schemes for data origin authentication, reducing the cost of conventional digital signature schemes, and designing fast authentication schemes. For every approach, we evaluated data origin authentication schemes from the associated classes and found that schemes from each class comprise a trade-off from a specific point of view.

We suggested a new class of data origin authentication schemes—unrestricted-time high-speed signing—that follows the latest approach of designing fast authentication schemes. In the unrestricted-time high-speed signing class every packet is signed independently with a high-performance signature scheme. In this way, we have revised the common assumption that signing every packet is impractical due to the high computational cost of conventional signature schemes. To validate that the unrestricted-time high-speed signing class proposed in this thesis is generically applicable, we analyzed a set of applications in critical infrastructures: clock synchronization, sensor data collection in Smart Grids, and group communication in 5G networks. We showed that recently proposed high-performance signature schemes are perfectly suitable as foundation for generally applicable data origin authentication schemes that provide computational efficiency, low communication overhead, as well as all other desired properties besides information-theoretical security, which is only provided by schemes that are not generally applicable. Given the results, we are confident that unrestricted-time high-speed signing can be the foundation of what may become a secure group communication protocol in the future—comparable to what TLS is to unicast communication today.

Subliminal Communication We analyzed a threat to information security that may become prevalent when data origin authentication schemes are used on a large scale with unrestricted-time high-speed signing proposed in this thesis: subliminal channels. We conducted a theoretical analysis of EdDSA and identified a way to inject a subliminal channel in EdDSA signatures. The efficiency of the subliminal channel is substantial (nearly 50 %). A further class of recent signatures that provides post-quantum security are MQ signatures, in which we discovered ways to inject subliminal

channels as well. We then validated existing countermeasures against subliminal channels and found that none of the countermeasures is viable in the context of group communication.

We furthermore introduced the ChainChannels scheme to establish a hidden C&C infrastructure to control a botnet over public blockchains by exploiting subliminal channels. ChainChannels is applicable to a wide variety of blockchains and can even be distributed over multiple blockchains to increase obfuscation. While we showed how blockchains can be used for botnet control, ChainChannels can be easily adjusted to any other use case for hidden multicast communication, where information should covertly be distributed to a set of receivers.

Secure Clock Synchronization Based on our theoretical evaluation of data origin authentication schemes for group communication, we found that most data origin authentication schemes are unsuitable to secure multicast clock synchronization. Even worse, TESLA, which was proposed in the IETF to secure broadcast messages in NTP and is highly rated by the P1588 Security Subcommittee for securing PTP, is susceptible to message delay attacks in the context of clock synchronization. When studying delay attacks, we discovered a fundamental limitation of clock synchronization protocols in adversarial settings: clock synchronization protocols can either be high-precision or secure against delay attacks. Based on this insight, we derived a formula for calculating the clock offset bounds that can be guaranteed for a particular system in adversarial settings.

We propose SecureTime, a set of security measures to secure multicast clock synchronization. At the heart of the security measures is an unrestricted-time high-speed signing scheme. The security measures furthermore entail a sequence number and session keys to prevent replay attacks as well as a novel set of countermeasures to mitigate delay attacks. We analyzed the security that SecureTime provides against every severe attack and provide bounds on the delays that can be introduced maliciously.

Acronyms

ARP Address Resolution Protocol. 38

ASA Algorithm-Substitution Attack. 35

ATM Asynchronous Transfer Mode. 12

BGP Border Gateway Protocol. 117

BiBa Bins and Balls. 32, 66, 68, 69, 72–74

BM-SC Broadcast Multicast - Service Centre. 43–45, 107, 108

C&C Command & Control. 15, 30, 36, 115, 124–131, 135

CCA Cyber Critical Asset. 110

CDN Content Delivery Network. 12

CFF Cover-Free Family. 58, 65

COTS Commercial Off-The-Shelf. 76

CSA Chained Stream Authentication. 68

DDoS Distributed DoS. 30

DNS Domain Name System. 12, 117, 137

DoS Denial of Service. 30, 33, 39–41, 47, 49, 56, 60, 67, 68, 77, 106, 136

EMSS Efficient Multi-chained Stream Signature. 62, 63, 72–74

eNB Evolved Node B. 43, 44, 107, 108

eSAIDA enhanced SAIDA. 65

ESP Encapsulating Security Payload. 88

GDOI Group Domain of Interpretation. 43

GPS Global Positioning System. 27, 85, 99

H2A Hybrid Hash-chaining scheme for adaptive source Authentication. 63

HFE Hidden Field Equations. 26, 122

HORS Hash to Abtain Random Subsets. 32, 66, 69, 72–74, 138

IDA Information Dispersal Algorithm. 64, 65, 137

IDS Intrusion Detection System. 40

IETF Internet Engineering Task Force. 33, 77, 135

LAN Local Area Network. 33

MAC Message Authentication Code. 20, 21, 23, 32, 51, 58, 67, 72–74, 77, 78
MBMS Multimedia Broadcast/Multicast Service. 44, 45, 106, 137
MCPTT Mission Critical Push to Talk. 42
mDNS Multicast DNS. 12
MITM Man in the Middle. 38, 40, 85, 92, 93, 122
MMOG Massively Multiplayer Online Game. 12
MQ Multivariate Quadratic. 15, 25, 26, 75, 76, 113, 122–124, 134, 137
MQQ MQ Quasigroups. 26, 123
MTK MBMS Traffic Key. 45, 106
MTS Multiple-Time Signature. 50–54, 57, 64, 66–74, 76, 109

NIC Network Interface Card. 35, 104
NTP Network Time Protocol. 8, 13, 26, 27, 33, 34, 40, 42, 77, 99–101, 115, 117, 118, 135
NTS Network Time Security. 33, 34, 77, 79, 101

OFB Output Feedback. 122
OSI Open Systems Interconnection. 12, 33
OTS One-Time Signature. 50–54, 57, 59–62, 65, 66, 68, 69, 71–74, 138
OWD One-Way Delay. 28, 29, 34, 79, 80, 83, 89–95, 97, 98

P2P Peer to Peer. 12, 30
PARM Pollution Attack Resistant Multicast authentication. 65
PDC Phasor Data Concentrator. 45, 110, 118
PMU Phasor Measurement Unit. 45, 110, 116, 118
PPS Pulse per Second. 27, 85, 99
PTP Precision Time Protocol. 6, 8, 13, 26–28, 32–34, 40, 77, 79, 81–85, 87–93, 97, 98, 101, 105, 135

RLH Receiver driven Layer Hash-chaining. 52, 62, 72–74
RNTI Radio Network Temporary Identifier. 45, 106
RTD Round-Trip Delay. 28, 34, 92–96

SA Subversion Attack. 35
SAIDA Signature Amortization using IDA. 64, 65, 72–74, 136
SAVe Stream Authentication scheme for Videos. 65
SC-PTM Single Cell-Point To Multipoint. 44, 45, 106
SETUP Secretly Embedded Trapdoor with Universal Protection. 35

TESLA Timed Efficient Stream Loss-Tolerant Authentication. 33, 67–70, 72–74, 77–79, 135

TFC Traffic Flow Confidentiality. 88

TLS Transport Layer Security. 12, 26, 43, 79, 112, 114, 116–118, 134

TLV Type Length Value. 105

TSV Tunable Signing and Verification. 69

TV-HORS Time Valid HORS. 32, 69, 76

TV-OTS Time-Valid OTS. 69

UE User Equipment. 43–45, 106–109

UOV Unbalanced Oil and Vinegar. 122, 123

UTC Coordinated Universal Time. 27, 85–87, 99

WAMPACS Wide Area Monitoring, Protection, and Control System. 13, 110

WAMS Wide Area Monitoring System. 110

Bibliography

3GPP. *TS 33.210 Network Domain Security (NDS); IP network layer security, Rel. 14*. Dec. 2016.

3GPP. *TS 33.246 Security of Multimedia Broadcast/Multicast Service (MBMS), Rel. 14*. Dec. 2016.

3GPP. *TS 33.401 3GPP System Architecture Evolution (SAE), Rel. 15*. June 2017.

Ali, Syed Taha, Patrick McCorry, Peter Hyun-Jeen Lee, and Feng Hao. "ZombieCoin 2.0: Managing next-Generation Botnets Using Bitcoin". In: *International Journal of Information Security* (June 2017), pp. 1–12.

Ali, Syed Taha, Patrick McCorry, Peter Hyun-Jeen Lee, and Feng Hao. "ZombieCoin: Powering Next-Generation Botnets with Bitcoin". In: *Financial Cryptography and Data Security*. Lecture Notes in Computer Science. Springer, Jan. 2015, pp. 34–48. ISBN: 978-3-662-48051-9.

Anderson, Ross, Serge Vaudenay, Bart Preneel, and Kaisa Nyberg. "The Newton channel". In: *Information Hiding: First International Workshop*. Springer, 1996, pp. 151–156. ISBN: 978-3-540-49589-5. DOI: 10.1007/3-540-61996-8_38.

Annessi, Robert, Joachim Fabini, Felix Iglesias, and Tanja Zseby. *Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization*. 2018. arXiv: 1811.08569 [cs.CR].

Annessi, Robert, Joachim Fabini, and Tanja Zseby. "It's about Time: Securing Broadcast Time Synchronization with Data Origin Authentication". In: *International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–11. DOI: 10.1109/ICCCN.2017.8038418.

Annessi, Robert, Joachim Fabini, and Tanja Zseby. *SecureTime: Secure Multicast Time Synchronization*. 2017. arXiv: 1705.10669 [cs.CR].

Annessi, Robert, Joachim Fabini, and Tanja Zseby. "To Trust or Not to Trust: Data Origin Authentication for Group Communication in 5G Networks". In: *International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: ACM, 2018, 43:1–43:7. ISBN: 978-1-4503-6448-5. DOI: 10.1145/3230833.3233252. URL: http://doi.acm.org/10.1145/3230833.3233252.

Annessi, Robert and Martin Schmiedecker. "NavigaTor: Finding Faster Paths to Anonymity". In: *IEEE European Symposium on Security and Privacy (Euro S&P)*. 2016. DOI: 10.1109/EuroSP.2016.26.

Annessi, Robert, Tanja Zseby, and Joachim Fabini. "A new Direction for Research on Data Origin Authentication in Group Communication". In: *International Conference on Cryptology and Network Security (CANS)*. Springer, 2017. DOI: 10.1007/978-3-030-02641-7_26.

Araniti, Giuseppe, Massimo Condoluci, Pasquale Scopelliti, Antonella Molinaro, and Antonio Iera. "Multicasting over Emerging 5G Networks: Challenges and Perspectives". In: *IEEE Network* 31.2 (Mar. 2017), pp. 80–89. ISSN: 0890-8044. DOI: 10.1109/MNET.2017.1600067NM.

Aslan, Heba K. "A hybrid scheme for multicast authentication over lossy networks". In: *Computers & Security* 23.8 (Dec. 2004), pp. 705–713. ISSN: 01674048. DOI: 10.1016/j.cose.2004.06.010.

Ateniese, Giuseppe, Bernardo Magri, and Daniele Venturi. "Subversion-Resilient Signature Schemes". In: *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. Denver, Colorado, USA, 2015, pp. 364–375. ISBN: 978-1-4503-3832-5.

Bellare, Mihir, Kenneth G. Paterson, and Phillip Rogaway. "Security of Symmetric Encryption against Mass Surveillance". In: *Advances in Cryptology*. Springer, 2014, pp. 1–19. ISBN: 978-3-662-44371-2. DOI: 10.1007/978-3-662-44371-2_1.

Bergadano, F., D. Cavagnino, and B. Crispo. "Individual single source authentication on the MBONE". In: *IEEE International Conference on Multimedia and Expo (ICME)*. 2000, pp. 541–544. DOI: 10.1109/ICME.2000.869659.

Bernstein, Daniel J. "Curve25519: New Diffie-Hellman Speed Records". In: *International Conference on Theory and Practice in Public-Key Cryptography (PKC)*. Springer, 2006, pp. 207–228. ISBN: 978-3-540-33852-9. DOI: 10.1007/11745853_14.

Bernstein, Daniel J., Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. "High-speed high-security signatures". In: *Journal of Cryptographic Engineering* 2.2 (2012), pp. 77–89.

Bitcoins the Hard Way: Using the Raw Bitcoin Protocol [Online]. http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html [Accessed: 18-Sep-2018].

Bohli, Jens-Matthias and Rainer Steinwandt. "On Subliminal Channels in Deterministic Signature Schemes". In: *International Conference on Information Security and Cryptology (ICISC)*. Springer, 2005, pp. 182–194. ISBN: 978-3-540-32083-8. DOI: 10.1007/11496618_14.

Bohli, Jens-Matthias, Maria Isabel Gonzalez Vasco, and Rainer Steinwandt. "A subliminal-free variant of ECDSA". In: *International Workshop on Information Hiding*. Springer, 2006, pp. 375–387.

Bos, Joppe W., J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig, and Eric Wustrow. "Elliptic Curve Cryptography in Practice". In: *Financial Cryptography and Data Security*. Lecture Notes in Computer Science. Springer, Mar. 2014, pp. 157–175. ISBN: 978-3-662-45472-5.

Brickell, Ernest F. "Some Ideal Secret Sharing Schemes". In: *Advances in Cryptology — EUROCRYPT '89*. Lecture Notes in Computer Science. Springer, Apr. 1989, pp. 468–475. ISBN: 978-3-540-46885-1.

Brown., Daniel R. L. "SEC 2: Recommended Elliptic Curve Domain Parameters (Version 2.0)". In: *Standards for Efficient Cryptography Group (SECG)* (Jan. 2010).

Canetti, R., J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. "Multicast Security: A Taxonomy and Some Efficient Constructions". In: Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). Vol. 2. Mar. 1999, pp. 708–716. DOI: 10.1109/INFCOM.1999.751457.

Carlén, P. "Traffic Flow Confidentiality mechanisms and their impact on traffic". In: *2013 Military Communications and Information Systems Conference*. Oct. 2013, pp. 1–6.

Challal, Y., H. Bettahar, and A. Bouabdallah. "A^2cast: an Adaptive source Authentication protocol for multiCAST streams". In: *International Symposium on Computers and Communications*. June 2004, 363–368 Vol.1. DOI: 10.1109/ISCC.2004.1358431.

Challal, Y., H. Bettahar, and A. Bouabdallah. "A taxonomy of multicast data origin authentication: Issues and solutions". In: *IEEE Communications Surveys Tutorials* 6.3 (2004), pp. 34–57. ISSN: 1553-877X. DOI: 10.1109/COMST.2004.5342292.

Challal, Yacine, Hatem Bettahar, and Abdelmadjid Bouabdallah. "Hybrid and Adaptive Hash-Chaining Scheme for Data-Streaming Source Authentication". In: *High Speed Networks and Multimedia Communications*. Lecture Notes in Computer Science 3079. Springer, 2004, pp. 1056–1067. ISBN: 978-3-540-25969-5.

Challal, Yacine and Abdelmadjid Bouabdallah. "Authenticast: a source authentication protocol for multicast flows and streams". In: *International Conference on Information Security*. 2005, pp. 175–178.

Challal, Yacine, Abdelmadjid Bouabdallah, and Hatem Bettahar. "H_2A: Hybrid Hash-chaining scheme for Adaptive multicast source authentication of media-streaming". In: *Computers & Security* 24.1 (Feb. 2005), pp. 57–68. ISSN: 0167-4048. DOI: 10.1016/j.cose.2004.06.012.

Challal, Yacine, Abdelmadjid Bouabdallah, and Yoann Hinard. "RLH: receiver driven layered hash-chaining for multicast data origin authentication". In: *Computer Communications* 28.7 (2005), pp. 726–740.

Chen, Ming-Shing, Bo-Yin Yang, and Daniel Smith-Tone. "PFLASH - Secure Asymmetric Signatures on Smart Cards". In: *Lightweight Cryptographic Workshop*. 2015.

Cherdantseva, Y. and J. Hilton. "A Reference Model of Information Assurance & Security". In: *2013 Eighth International Conference on Availability, Reliability and Security (ARES)*. Sept. 2013, pp. 546–555. DOI: 10.1109/ARES.2013.72.

Council of European Union. *Directive 2014/65/EU of the european parliament and of the council of 15 May 2014 on markets in financial instruments and amending Directive 2002/92/EC and Directive 2011/61/EU*. https://eur-lex.europa.eu/eli/dir/2014/65/oj. 2014.

Courtois, Nicolas T, Louis Goubin, and Jacques Patarin. "SFLASHv3, a fast asymmetric signature scheme." In: *IACR Cryptology ePrint Archive* 2003 (2003), p. 211.

Courtois, Nicolas, Louis Goubin, and Jacques Patarin. "Quartz, an asymmetric signature scheme for short signatures on PC Primitive specification and supporting documentation". In: (2001).

Cucinotta, Tommaso, Gabriele Cecchetti, and Gianluca Ferraro. "Adopting redundancy techniques for multicast stream authentication". In: *IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS)*. 2003.

Czosseck, C., G. Klein, and F. Leder. "On the Arms Race around Botnets - Setting up and Taking down Botnets". In: *International Conference on Cyber Conflict*. June 2011, pp. 1–14.

Decker, Christian and Roger Wattenhofer. "Bitcoin Transaction Malleability and MtGox". In: *Computer Security - ESORICS*. Lecture Notes in Computer Science. Springer, Sept. 2014, pp. 313–326. ISBN: 978-3-319-11212-1.

Desmedt, Y., Y. Frankel, and M. Yung. "Multi-receiver/multi-sender network security: efficient authenticated multicast/feedback". In: *Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM)*. May 1992, 2045–2054 vol.3. DOI: 10.1109/INFCOM.1992.263476.

Desmedt, Yvo and Goce Jakimoski. "Non-degrading Erasure-Tolerant Information Authentication with an Application to Multicast Stream Authentication over Lossy Channels". In: *Topics in Cryptology – (CT-RSA)*. Lecture Notes in Computer Science 4377. Springer, Feb. 5, 2007, pp. 324–338. ISBN: 978-3-540-69328-4.

Diffie, Whitfield and Martin E. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654.

Ding, Jintai and Dieter Schmidt. "Rainbow, a New Multivariable Polynomial Signature Scheme". In: *International Conference on Applied Cryptography and Network Security, ACNS 2005*. Springer, 2005, pp. 164–175. ISBN: 978-3-540-31542-1.

Dolev, Danny and Andrew C Yao. "On the security of public key protocols". In: *IEEE Transactions on Information Theory* 29.2 (1983), pp. 198–208.

Dong, Q. and G. Xiao. "A Subliminal-Free Variant of ECDSA Using Interactive Protocol". In: *International Conference on E-Product E-Service and E-Entertainment*. Nov. 2010, pp. 1–3. DOI: 10.1109/ICEEE.2010.5660874.

Dowling, Benjamin, Douglas Stebila, and Greg Zaverucha. "Authenticated Network Time Synchronization". In: *USENIX Security Symposium*. Austin, TX, Aug. 2016, pp. 823–840. ISBN: 978-1-931971-32-4.

ElGamal, Taher. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *Advances in cryptology*. Springer. 1985, pp. 10–18.

Ellegaard, L. *PTP Security using MACsec*. Tech. rep. https://ieee-sa.centraldesktop.com/1588/file/33390811/. P1588 Working Group, Aug. 2014.

European Commision. *Annex to the Commission delegated regulation supplementing Directive 2014/65/EU of the European Parliament and of the Council with regard to regulatory technical standards for the level of accuracy of business clocks*. http://ec.europa.eu/finance/securities/docs/isd/mifid/rts/160607-rts-25-annex_en.pdf. 2016.

European Commision. *Commission delegated regulation (EU) of 7.6.2016 supplementing Directive 2014/65/EU of the European Parliament and of the Council with regard to regulatory technical standards for the level of accuracy of business clocks*. http://ec.europa.eu/finance/securities/docs/isd/mifid/rts/160607-rts-25_en.pdf. 2016.

Even, Shimon, Oded Goldreich, and Silvio Micali. "On-line/off-line digital signatures". In: *Journal of Cryptology* 9.1 (1996), pp. 35–67.

Faugere, Jean-Charles, Danilo Gligoroski, Ludovic Perret, Simona Samardjiska, and Enrico Thomae. "A polynomial-time key-recovery attack on MQQ cryptosystems". In: *IACR International Workshop on Public Key Cryptography*. Springer, 2015, pp. 150–174.

FIPS, NIST. "198: The keyed-hash message authentication code (HMAC)". In: *National Institute of Standards and Technology, Federal Information Processing Standards* (2002), p. 29.

Frkat, Davor, Robert Annessi, and Tanja Zseby. "ChainChannels: Private Botnet Communication Over Public Blockchains". In: *IEEE International Conference on Blockchain (Blockchain)*. 2018.

Fujii, Hiroshi, Wattanawong Kachen, and Kaoru Kurosawa. "Combinatorial bounds and design of broadcast authentication". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 79.4 (1996), pp. 502–506.

Gao, Chong-zhi and Zheng-an Yao. "How to Authenticate Real Time Streams Using Improved Online/Offline Signatures". In: *Cryptology and Network Security*. Lecture Notes in Computer Science 3810. Springer, 2005, pp. 134–146. ISBN: 978-3-540-32298-6.

Gennaro, Rosario and Pankaj Rohatgi. "How to Sign Digital Streams". In: *Information and Computation* 165.1 (2001), pp. 100–116. ISSN: 0890-5401. DOI: 10.1006/inco.2000.2916.

Gennaro, Rosario and Pankaj Rohatgi. "How to sign digital streams". In: vol. Advances in Cryptology. Springer, 1997, pp. 180–197.

Gilbert, Edgar N, F Jessie MacWilliams, and Neil JA Sloane. "Codes which detect deception". In: *Bell System Technical Journal* 53.3 (1974), pp. 405–424.

Gligoroski, Danilo, Smile Markovski, and Svein J. Knapskog. "A Public Key Block Cipher Based on Multivariate Quadratic Quasigroups". In: *IACR Cryptology ePrint Archive* (2008). URL: http://eprint.iacr.org/2008/320.pdf.

Gligoroski, Danilo, Smile Markovski, and Svein Johan Knapskog. "Multivariate Quadratic Trapdoor Functions Based on Multivariate Quadratic Quasigroups". In: *American Conference on Applied Mathematics*. MATH. Cambridge, Massachusetts: WSEAS, 2008, pp. 44–49. ISBN: 978-960-6766-47-3.

Gligoroski, Danilo et al. "MQQ-SIG". In: *Trusted Systems*. Springer, 2011, pp. 184–203.

Gligoroski, Danilo et al. "MQQ-SIG: An Ultra-fast and Provably CMA Resistant Digital Signature Scheme". In: *International Conference on Trusted Systems*. INTRUST'11. Beijing, China: Springer, 2012, pp. 184–203. ISBN: 978-3-642-32297-6.

Goldreich, Oded. *Foundations of Cryptography: Volume 2, Basic Applications*. USA: Cambridge University Press, 2004. ISBN: 978-0-521-83084-3.

Golle, Philippe and Nagendra Modadugu. "Authenticating Streamed Data in the Presence of Random Packet Loss." In: *NDSS*. 2001, pp. 13–22.

Grover, Kanika and Alvin Lim. "A survey of broadcast authentication schemes for wireless networks". In: *Ad Hoc Networks* 24 (Jan. 2015), pp. 288–316. ISSN: 15708705. DOI: 10.1016/j.adhoc.2014.06.008.

Haller, Neil. "The S/KEY One-Time Password System". In: *ISOC Symposium on Network and Distributed System Security*. San Diego, CA, Feb. 1994.

Hamburg, Mike. *Ed448-Goldilocks, a new elliptic curve*. Cryptology ePrint Archive, Report 2015/625. http://eprint.iacr.org/2015/625. 2015.

Hardjono, Thomas, Lakshminath R. Dondeti, and Radia Perlman. *Multicast and Group Security*. USA: Artech House, Inc., 2003. ISBN: 1580533426.

Hardjono, Thomas and Gene Tsudik. "IP multicast security: Issues and directions". In: *Annales des télécommunications*. Vol. 55. 7-8. Springer. 2000, pp. 324–340.

Hartl, Alexander, Robert Annessi, and Tanja Zseby. "A Subliminal Channel in EdDSA: Information Leakage with High-Speed Signatures". In: *International Workshop on Managing Insider Security Threats*. MIST '17. Dallas, Texas, USA: ACM, 2017, pp. 67–78. ISBN: 978-1-4503-5177-5. DOI: 10.1145/3139923.3139925.

Hartl, Alexander, Robert Annessi, and Tanja Zseby. "Subliminal Channels in High-Speed Signatures". In: *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)* 9.1 (Mar. 2018), pp. 30–53.

He, Jinxin, Gaochao Xu, Xiaodong Fu, Zhiguo Zhou, and Jianhua Jiang. "Survey on multicast data origin authentication". In: *IEEE International Conference on Communication Technology (ICCT)*. Nov. 2008, pp. 749–752. DOI: 10.1109/ICCT.2008.4716234.

IEC/TR 61850-90-1:2010. *Communication networks and systems for power utility automation – Part 90-1: Use of IEC 61850 for the communication between substations*. IEC, Geneva, Switzerland, 2010.

IEC/TS 62351-1:2007. *Power systems management and associated information exchange - Data and communications security – Part 1: Communication network and system security - Introduction to security issues*. IEC, Geneva, Switzerland, 2007.

"IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads". In: *IEEE Std 2030-2011* (Sept. 2011), pp. 1–126. DOI: 10.1109/IEEESTD.2011.6018239.

"IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems". In: *IEEE Std 1588-2008* (July 2008), pp. 1–269. DOI: 10.1109/IEEESTD.2008.4579760.

"IEEE Standard for Synchrophasor Measurements for Power Systems". In: *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)* (Dec. 2011), pp. 1–61. DOI: 10.1109/IEEESTD.2011.6111219.

Iglesias, Felix, Valentin Bernhardt, Robert Annessi, and Tanja Zseby. "Decision Tree Rule Induction for Detecting Covert Timing Channels in TCP/IP Traffic". In: *International Cross Domain Conference for Machine Learning & Knowledge Extraction (CD-MAKE)*. 2017.

Imai, Hideki and Tsutomu Matsumoto. "Algebraic methods for constructing asymmetric cryptosystems". In: *International Conference on Algebraic Algorithms and Error-Correcting Codes*. Springer, 1986, pp. 108–119. ISBN: 978-3-540-39855-4. DOI: 10.1007/3-540-16776-5_713.

Itkin, E. and A. Wool. "A security analysis and revised security extension for the precision time protocol". In: *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. Sept. 2016, pp. 1–6. DOI: 10.1109/ISPCS.2016.7579501.

Itkin, Eyal and Avishai Wool. "A Security Analysis and Revised Security Extension for the Precision Time Protocol". In: *CoRR* abs/1603.00707 (May 28, 2016). URL: http://arxiv.org/abs/1603.00707.

Jakimoski, Goce. "Unconditionally Secure Information Authentication in Presence of Erasures". In: *Cryptography and Coding*. Lecture Notes in Computer Science 3796. Springer, Dec. 19, 2005, pp. 304–321. ISBN: 978-3-540-32418-8.

Jeong, JaeYong, Yongsu Park, and Yookun Cho. "Efficient DoS Resistant Multicast Authentication Schemes". In: *Computational Science and Its Applications – ICCSA*. Lecture Notes in Computer Science 3481. Springer, 2005, pp. 353–362. ISBN: 978-3-540-32044-9.

Johnson, Don, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)". In: *International Journal of Information Security* (2001), pp. 36–63.

Judge, P. and M. Ammar. "Security issues and solutions in multicast content distribution: a survey". In: *IEEE Network* 17.1 (Jan. 2003), pp. 30–36. ISSN: 0890-8044. DOI: 10.1109/MNET.2003.1174175.

Katz, Jonathan. *Digital Signatures*. Boston, MA: Springer US, 2010. ISBN: 978-0-387-27711-0.

Kaur, Ramanpreet, Amrit Lal Sangal, and Krishan Kumar. "Mac based multicast source authentication: A survey". In: *International Journal of Computer Application* 37.2 (2012).

Kim, J., S. W. Choi, W. Y. Shin, Y. S. Song, and Y. K. Kim. "Group communication over LTE: a radio access perspective". In: *IEEE Communications Magazine* 54.4 (Apr. 2016), pp. 16–23. ISSN: 0163-6804. DOI: 10.1109/MCOM.2016.7452261.

Kipnis, Aviad, Jacques Patarin, and Louis Goubin. "Unbalanced Oil and Vinegar Signature Schemes". In: *Advances in Cryptology — EUROCRYPT*. Springer, 1999, pp. 206–222. ISBN: 978-3-540-48910-8. DOI: 10.1007/3-540-48910-X_15.

Koskiahde, T., J. Kujala, and T. Norolampi. "A sensor network architecture for military and crisis management". In: *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. Sept. 2008, pp. 110–114. DOI: 10.1109/ISPCS.2008.4659223.

Kurosawa, Kaoru and Satoshi Obana. "Characterization of (k, n) multi-receiver authentication". In: *Information Security and Privacy*. Lecture Notes in Computer Science 1270. Springer, July 7, 1997, pp. 204–215. ISBN: 978-3-540-69237-9.

Lamport, Leslie. *Constructing digital signatures from a one-way function*. Technical Report CSL-98, SRI International Palo Alto, 1979.

Lamport, Leslie. "Password authentication with insecure communication". In: *Communications of the ACM* 24.11 (1981), pp. 770–772.

Law, Yee Wei, Zheng Gong, Tie Luo, Slaven Marusic, and Marimuthu Palaniswami. "Comparative Study of Multicast Authentication Schemes with Application to Wide-area Measurement System". In: *ACM SIGSAC Symposium on Information, Computer and Communications Security*. ASIACCS. 2013, pp. 287–298. ISBN: 978-1-4503-1767-2. DOI: 10.1145/2484313.2484349.

Law, Yee Wei, Marimuthu Palaniswami, Gina Kounga, and Anthony Lo. "WAKE: Key management scheme for wide-area measurement systems in smart grid". In: *Communications Magazine, IEEE* 51.1 (2013), pp. 34–41.

Lee, Jaeheung, Seokhyun Kim, Yookun Cho, Yoojin Chung, and Yongsu Park. "HORSIC: An efficient one-time signature scheme for wireless sensor networks". In: *Information Processing Letters* 112.20 (Oct. 31, 2012), pp. 783–787. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2012.07.007.

Li, Qing and W. Trappe. "Staggered TESLA: a multicast authentication scheme resistant to DoS attacks". In: *IEEE Global Telecommunications Conference (GLOBECOM)*. Nov. 2005. DOI: 10.1109/GLOCOM.2005.1577934.

Li, Qinghua and Guohong Cao. "Multicast Authentication in the Smart Grid With One-Time Signature". In: *IEEE Transactions on Smart Grid* 2.4 (Dec. 2011), pp. 686–696. ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2138172.

Lin, Ya-Jeng, Shiuhpyng Shieh, and Warren W. Lin. "Lightweight, Pollution-attack Resistant Multicast Authentication Scheme". In: *ACM Symposium on Information, Computer and Communications Security*. ASIACCS. USA, 2006, pp. 148–156. ISBN: 978-1-59593-272-3. DOI: 10.1145/1128817.1128840.

Lisova, Elena et al. "Protecting Clock Synchronization: Adversary Detection through Network Monitoring". In: *Journal of Electrical and Computer Engineering* 2016 (2016), pp. 1–13. ISSN: 2090-0147, 2090-0155. DOI: 10.1155/2016/6297476. URL: http://www.hindawi.com/journals/jece/2016/6297476/ (visited on 02/08/2017).

Liu, D., Peng Ning, Sencun Zhu, and S. Jajodia. "Practical broadcast authentication in sensor networks". In: *International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*. July 2005, pp. 118–129. DOI: 10.1109/MOBIQUITOUS.2005.49.

Liu, Donggang and Peng Ning. "Multilevel μTESLA: Broadcast Authentication for Distributed Sensor Networks". In: *ACM Transactions Embedded Computing Systems* 3.4 (Nov. 2004), pp. 800–836. ISSN: 1539-9087. DOI: 10.1145/1027794.1027800.

Malhotra, Aanchal and Sharon Goldberg. "Attacking NTP's Authenticated Broadcast Mode". In: *ACM SIGCOMM Computer Communication Review* 46.1 (2016), pp. 12–17.

Mark L. Psiaki and Todd E. Humphreys. *Protecting GPS From Spoofers Is Critical to the Future of Navigation*. https://spectrum.ieee.org/telecom/security/protecting-gps-from-spoofers-is-critical-to-the-future-of-navigation. 2016.

Matsumoto, Tsutomu and Hideki Imai. "Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption". In: *Advances in Cryptology — EUROCRYPT: Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1988, pp. 419–453. ISBN: 978-3-540-45961-3. DOI: 10.1007/3-540-45961-8_39.

Matzutt, Roman, Jens Hiller, Martin Henze, and Jan Henrik Ziegeldorf. "A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin". In: *International Conference on Financial Cryptography and Data Security*. Springer, 2018, p. 18.

Menezes, Alfred J, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

Merkle, Ralph. *Comments in 2012 about the 1979 paper: A Certified Digital Signature*. http://www.merkle.com/papers/Certified1979.pdf. [Online; accessed 08-August-2018]. 2012.

Merkle, Ralph C. "A certified digital signature". In: *Advances in Cryptology—CRYPTO*. Springer, 1989, pp. 218–238.

Mills, David L. *Computer Network Time Synchronization: the Network Time Protocol on Earth and in Space*. 2nd ed. CRC Press, 2011. ISBN: 978-1-4398-1463-5.

Miner, Sara and Jessica Staddon. "Graph-based authentication of digital streams". In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2001, pp. 232–246.

Mitzenmacher, Michael D and Adrian Perrig. "Bounds and improvements for BiBa signature schemes". In: (2002).

Mizrahi, T. "Time synchronization security using IPsec and MACsec". In: *2011 International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*. Sept. 2011, pp. 38–43. DOI: 10.1109/ISPCS.2011.6070153.

Mizrahi, Tal. "A game theoretic analysis of delay attacks against time synchronization protocols". In: *International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*. IEEE, 2012, pp. 1–6.

Moeser, M., R. Boehme, and D. Breuker. "An Inquiry into Money Laundering Tools in the Bitcoin Ecosystem". In: *2013 APWG eCrime Researchers Summit*. Sept. 2013, pp. 1–14.

Moreira, N., J. Lázaro, J. Jimenez, M. Idirin, and A. Astarloa. "Security mechanisms to protect IEEE 1588 synchronization: State of the art and trends". In: IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS). Oct. 2015, pp. 115–120. DOI: 10.1109/ISPCS.2015.7324694.

Morris, T.H., Shengyi Pan, and U. Adhikari. "Cyber security recommendations for wide area monitoring, protection, and control systems". In: *IEEE Power and Energy Society General Meeting*. July 2012, pp. 1–6. DOI: 10.1109/PESGM.2012.6345127.

Moyer, M.J., J.R. Rao, and P. Rohatgi. "A survey of security issues in multicast communications". In: *IEEE Network* 13.6 (Nov. 1999), pp. 12–23. ISSN: 0890-8044. DOI: 10.1109/65.806981.

Nadji, Y., R. Perdisci, and M. Antonakakis. "Still Beheading Hydras: Botnet Takedowns Then and Now". In: *IEEE Transactions on Dependable and Secure Computing* 14.5 (Sept. 2017), pp. 535–549.

Naor, M. and O. Reingold. "Number-theoretic constructions of efficient pseudo-random functions". In: *Symposium on Foundations of Computer Science*. Oct. 1997, pp. 458–467. DOI: 10.1109/SFCS.1997.646134.

Narayanan, Arvind, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, July 2016. ISBN: 978-0-691-17169-2.

NIST, SP. "800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication". In: *NIST Special Publication* (2005).

Obana, Satoshi and Kaoru Kurosawa. "Bounds and Combinatorial Structure of (k,n) Multi-Receiver A-Codes". In: *Designs, Codes and Cryptography* 22.1 (Jan. 2001), pp. 47–63. ISSN: 0925-1022, 1573-7586. DOI: 10.1023/A:1008351225940.

Pannetrat, Alain and Refik Molva. "Efficient Multicast Packet Authentication." In: *NDSS*. 2003.

Pannetrat, Alain and Réfik Molva. "Authenticating real time packet streams and multicasts". In: *International Symposium on Computers and Communications, ISCC*. IEEE, 2002, pp. 490–495.

Park, Jung Min, Edwin KP Chong, and Howard Jay Siegel. "Efficient multicast packet authentication using signature amortization". In: *IEEE Symposium on Security and Privacy*. IEEE, 2002, pp. 227–240.

Park, Jung Min, Edwin KP Chong, and Howard Jay Siegel. "Efficient multicast stream authentication using erasure codes". In: *ACM Transactions on Information and System Security (TISSEC)* 6.2 (2003), pp. 258–285.

Park, Yongsu and Yookun Cho. "Efficient one-time signature schemes for stream authentication". In: *Journal of Information Science and Engineering* (2006).

Park, Yongsu and Yookun Cho. "The eSAIDA Stream Authentication Scheme". In: *Computational Science and Its Applications – ICCSA*. Lecture Notes in Computer Science 3046. Springer, May 14, 2004, pp. 799–807. ISBN: 978-3-540-24768-5.

Patarin, Jacques. "Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms". In: *Advances in Cryptology — EUROCRYPT*. Springer, 1996, pp. 33–48. ISBN: 978-3-540-68339-1.

Patarin, Jacques. "The oil and vinegar signature scheme". In: *Dagstuhl Workshop on Cryptography*. 1997.

Perrig, Adrian. "The BiBa One-time Signature and Broadcast Authentication Protocol". In: *ACM Conference on Computer and Communications Security*. CCS. New York, USA: ACM, 2001, pp. 28–37. ISBN: 1-58113-385-5. DOI: 10.1145/501983.501988.

Perrig, Adrian, Ran Canetti, Dawn Song, and J. Doug Tygar. "Efficient and Secure Source Authentication for Multicast". In: *Network and Distributed System Security Symposium, NDSS*. 2001, pp. 35–46.

Perrig, Adrian, Ran Canetti, J. Doug Tygar, and Dawn Song. "Efficient authentication and signing of multicast streams over lossy channels". In: *IEEE Symposium on Security and Privacy (S&P)*. 2000, pp. 56–73.

Perrig, Adrian, Ran Canetti, J. Doug Tygar, and Dawn Song. "The TESLA broadcast authentication protocol". In: *RSA CryptoBytes* 5 (2002).

Perrig, Adrian, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. "SPINS: Security Protocols for Sensor Networks". In: *Wireless Networks*. 2001, pp. 189–199.

Perrig, Adrian and J. D. Tygar. *Secure Broadcast Communication*. Springer, 2003. ISBN: 978-1-4615-0229-6.

Petzoldt, Albrecht, Ming-Shing Chen, Bo-Yin Yang, Chengdong Tao, and Jintai Ding. "Design Principles for HFEv-Based Multivariate Signature Schemes". In: *Advances in Cryptology – ASIACRYPT*. Springer, 2015, pp. 311–334. ISBN: 978-3-662-48797-6.

Pieprzyk, Josef, Jennifer Seberry, and Thomas Hardjono. *Fundamentals of Computer Security*. USA: Springer, 2002. ISBN: 3540431012.

Popescu, A., D. Constantinescu, D. Erman, and D. Ilie. "A Survey of Reliable Multicast Communication". In: *Next Generation Internet Networks*. May 2007, pp. 111–118. DOI: 10.1109/NGI.2007.371205.

Rabin, Michael O. *Digitalized signatures and public-key functions as intractable as factorization*. LCS TR-212. MIT, 1979.

Rabin, Michael O. "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance". In: *J. ACM* 36.2 (Apr. 1989), pp. 335–348. ISSN: 0004-5411. DOI: 10.1145/62044.62050.

Reyzin, Leonid and Natan Reyzin. "Better than BiBa: Short one-time signatures with fast signing and verifying". In: *Information Security and Privacy*. Springer, 2002, pp. 144–153.

Haller, N. *The S/KEY One-Time Password System*. RFC 1760 (Informational). RFC. Fremont, CA, USA: RFC Editor, Feb. 1995. DOI: 10.17487/RFC1760. URL: https://www.rfc-editor.org/rfc/rfc1760.txt.

Deering, S. and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460 (Draft Standard). RFC. Obsoleted by RFC 8200, updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946, 7045, 7112. Fremont, CA, USA: RFC Editor, Dec. 1998. DOI: 10.17487/RFC2460. URL: https://www.rfc-editor.org/rfc/rfc2460.txt.

Perrig, A., D. Song, R. Canetti, J. D. Tygar, and B. Briscoe. *Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction*. RFC 4082 (Informational). RFC. Fremont, CA, USA: RFC Editor, June 2005. DOI: 10.17487/RFC4082. URL: https://www.rfc-editor.org/rfc/rfc4082.txt.

Kent, S. *IP Encapsulating Security Payload (ESP)*. RFC 4303 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Dec. 2005. DOI: 10.17487/RFC4303. URL: https://www.rfc-editor.org/rfc/rfc4303.txt.

Shirey, R. *Internet Security Glossary, Version 2*. RFC 4949 (Informational). RFC. Fremont, CA, USA: RFC Editor, Aug. 2007. DOI: 10.17487/RFC4949. URL: https://www.rfc-editor.org/rfc/rfc4949.txt.

Dierks, T. and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246 (Proposed Standard). RFC. Obsoleted by RFC 8446, updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685, 7905, 7919, 8447. Fremont, CA, USA: RFC Editor, Aug. 2008. DOI: 10.17487/RFC5246. URL: https://www.rfc-editor.org/rfc/rfc5246.txt.

Mills, D., J. Martin (Ed.), J. Burbank, and W. Kasch. *Network Time Protocol Version 4: Protocol and Algorithms Specification*. RFC 5905 (Proposed Standard). RFC. Updated by RFC 7822. Fremont, CA, USA: RFC Editor, June 2010. DOI: 10.17487/RFC5905. URL: https://www.rfc-editor.org/rfc/rfc5905.txt.

Haberman (Ed.), B. and D. Mills. *Network Time Protocol Version 4: Autokey Specification*. RFC 5906 (Informational). RFC. Fremont, CA, USA: RFC Editor, June 2010. DOI: 10.17487/RFC5906. URL: https://www.rfc-editor.org/rfc/rfc5906.txt.

Frankel, S. and S. Krishnan. *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*. RFC 6071 (Informational). RFC. Fremont, CA, USA: RFC Editor, Feb. 2011. DOI: 10.17487/RFC6071. URL: https://www.rfc-editor.org/rfc/rfc6071.txt.

Rescorla, E. and N. Modadugu. *Datagram Transport Layer Security Version 1.2*. RFC 6347 (Proposed Standard). RFC. Updated by RFCs 7507, 7905. Fremont, CA, USA: RFC Editor, Jan. 2012. DOI: 10.17487/RFC6347. URL: https://www.rfc-editor.org/rfc/rfc6347.txt.

Weis, B., S. Rowles, and T. Hardjono. *The Group Domain of Interpretation*. RFC 6407 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Oct. 2011. DOI: 10.17487/RFC6407. URL: https://www.rfc-editor.org/rfc/rfc6407.txt.

Pornin, T. *Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)*. RFC 6979 (Informational). RFC. Fremont, CA, USA: RFC Editor, Aug. 2013. DOI: 10.17487/RFC6979. URL: https://www.rfc-editor.org/rfc/rfc6979.txt.

Mizrahi, T. *Security Requirements of Time Protocols in Packet Switched Networks*. RFC 7384 (Informational). RFC. Fremont, CA, USA: RFC Editor, Oct. 2014. DOI: 10.17487/RFC7384. URL: https://www.rfc-editor.org/rfc/rfc7384.txt.

Josefsson, S. and I. Liusvaara. *Edwards-Curve Digital Signature Algorithm (EdDSA)*. RFC 8032 (Informational). RFC. Fremont, CA, USA: RFC Editor, Jan. 2017. DOI: 10.17487/RFC8032. URL: https://www.rfc-editor.org/rfc/rfc8032.txt.

Sury, O. and R. Edmonds. *Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC*. RFC 8080 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Feb. 2017. DOI: 10.17487/RFC8080. URL: https://www.rfc-editor.org/rfc/rfc8080.txt.

Mattsson, J. and D. Migault. *ECDHE_PSK with AES-GCM and AES-CCM Cipher Suites for TLS 1.2 and DTLS 1.2*. RFC 8442 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Sept. 2018. DOI: 10.17487/RFC8442. URL: https://www.rfc-editor.org/rfc/rfc8442.txt.

Rescorla, E. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Aug. 2018. DOI: 10.17487/RFC8446. URL: https://www.rfc-editor.org/rfc/rfc8446.txt.

Rivest, Ronald L., Adi Shamir, and Len Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

Rogaway, Phillip and Thomas Shrimpton. "Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance". In: *Fast Software Encryption*. 3017. Springer, Feb. 5, 2004, pp. 371–388. ISBN: 978-3-540-25937-4. DOI: 10.1007/978-3-540-25937-4_24.

Rohatgi, Pankaj. "A Compact and Fast Hybrid Signature Scheme for Multicast Packet Authentication". In: *ACM Conference on Computer and Communications Security*. CCS. USA: ACM, 1999, pp. 93–100. ISBN: 1-58113-148-8. DOI: 10.1145/319709.319722.

Röttger, Stephen. "Analysis of the NTP Autokey Procedures". Master's thesis, Technische Universitt Braunschweig, 2012.

Safavi-Naini, R. and H. Wang. "Multireceiver Authentication Codes: Models, Bounds, Constructions, and Extensions". In: *Information and Computation* 151.1–2 (May 25, 1999), pp. 148–172. ISSN: 0890-5401. DOI: 10.1006/inco.1998.2769.

Safavi-Naini, R. and H. Wang. "New results on multi-receiver authentication codes". In: *Advances in Cryptology — (EUROCRYPT)*. Lecture Notes in Computer Science 1403. Springer, May 31, 1998, pp. 527–541. ISBN: 978-3-540-69795-4.

Sato, M. and S. Matsuo. "Long-Term Public Blockchain: Resilience against Compromise of Underlying Cryptography". In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. July 2017, pp. 1–8.

Schneier, Bruce. *Applied Cryptography: Protocols, Algorithms and Source Code in C 20th Anniversary Edition*. John Wiley & Sons Inc, May 2015. ISBN: 978-1-119-09672-6.

Schnorr, Claus P. "Efficient Identification and Signatures for Smart Cards". In: *Advances in Cryptology - CRYPTO*. Springer, 1990, pp. 239–252.

Seys, Stefaan and Bart Preneel. "Power consumption evaluation of efficient digital signature schemes for low power devices". In: *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications (WiMob)*. Vol. 1. IEEE, 2005, pp. 79–86.

Shamir, Adi. "How to Share a Secret". In: *Commun. ACM* 22.11 (Nov. 1979), pp. 612–613.

Shamir, Adi and Yael Tauman. "Improved online/offline signature schemes". In: *Advances in Cryptology*. Springer, 2001, pp. 355–367.

Shor, P.W. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Symposium on Foundations of Computer Science*. Nov. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

Shor, Peter W. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM J. Comput.* 26.5 (Oct. 1997), pp. 1484–1509. ISSN: 0097-5397. DOI: 10.1137/S0097539795293172. URL: http://dx.doi.org/10.1137/S0097539795293172.

Sibold, Dieter, Stephen Roettger, and Kristof Teichel. *Network Time Security*. Internet-Draft draft-ietf-ntp-network-time-security-11. Oct. 2015. URL: http://www.ietf.org/internet-drafts/draft-ietf-ntp-network-time-security-11.txt.

Sibold, Dieter, Stephen Roettger, and Kristof Teichel. *Using the Network Time Security Specification to Secure the Network Time Protocol*. Internet-Draft draft-ietf-ntp-using-nts-for-ntp-02. Oct. 2015. URL: http://www.ietf.org/internet-drafts/draft-ietf-ntp-using-nts-for-ntp-02.txt.

Sibold, Dieter, Kristof Teichel, Stephen Roettger, and Russ Housley. *Protecting Network Time Security Messages with the Cryptographic Message Syntax (CMS)*. Internet-Draft draft-ietf-ntp-cms-for-nts-message-06. Feb. 2016. URL: https://tools.ietf.org/html/draft-ietf-ntp-cms-for-nts-message-06.

Simmons, G. J. "A survey of information authentication". In: *Proceedings of the IEEE* 76.5 (May 1988), pp. 603–620. ISSN: 0018-9219. DOI: 10.1109/5.4445.

Simmons, Gustavus J. "Subliminal Communication Is Easy Using the DSA". In: *Advances in Cryptology — EUROCRYPT*. Lecture Notes in Computer Science. Springer, May 1993, pp. 218–232. ISBN: 978-3-540-48285-7.

Simmons, Gustavus J. "The Prisoners' Problem and the Subliminal Channel". In: *Advances in Cryptology*. DOI: 10.1007/978-1-4684-4730-9_5. Springer, 1984, pp. 51–67. ISBN: 978-1-4684-4730-9.

Steinwandt, Rainer and Viktória I. Villányi. "A One-time Signature Using Run-length Encoding". In: *Information Processing Letters* 108.4 (Oct. 2008), pp. 179–185. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2008.05.004.

Stenn, Harlan. *[ntpwg] Antw: Re: Proposed REFID changes*. http://lists.ntp.org/pipermail/ntpwg/2015-July/002291.html. [Online; accessed 12-September-2018]. 2015.

Sward, Andrew, Ivy Vecna, and Forrest Stonedahl. "Data Insertion in Bitcoin's Blockchain". In: *Ledger* 3.0 (Apr. 2018).

Tartary, C., Huaxiong Wang, and San Ling. "Authentication of Digital Streams". In: *IEEE Transactions on Information Theory* 57.9 (Sept. 2011), pp. 6285–6303. ISSN: 0018-9448. DOI: 10.1109/TIT.2011.2161960.

Tartary, Christophe Maurice Andre. *Authentication for Multicast Communication*. Macquarie University, 2007.

Teichel, Kristof, Dieter Sibold, and Stefan Milius. "An attack possibility on time synchronization protocols secured with TESLA-like mechanisms". In: *Information Systems Security*. Springer, 2016, pp. 3–22.

Teichel, Kristof, Dieter Sibold, and Stefan Milius. "First Results of a Formal Analysis of the Network Time Security Specification". In: *Security Standardisation Research*. Lecture Notes in Computer Science 9497. Springer, 2015, pp. 218–245. ISBN: 978-3-319-27152-1.

Telecommunication standardization sector of ITU. *Time and phase synchronization aspects of telecommunication networks*. https://www.itu.int/rec/T-REC-G.8271-201708-I. 2017.

Tesfay, Teklemariam and Jean-Yves Le Boudec. "Experimental Comparison of Multicast Authentication for Wide Area Monitoring Systems". In: *IEEE Transactions on Smart Grid* (2017). ISSN: 1949-3053, 1949-3061. DOI: 10.1109/TSG.2017.2656067.

The Smart Grid Interoperability Panel–Smart Grid Cybersecurity Committee. *Guidelines for Smart Grid Cybersecurity*. NIST IR 7628r1. National Institute of Standards and Technology, Sept. 2014.

Treytl, A. and B. Hirschler. "Securing IEEE 1588 by IPsec tunnels - An analysis". In: *International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*. Sept. 2010, pp. 83–90. DOI: 10.1109/ISPCS.2010.5609765.

Treytl, A. and B. Hirschler. "Security flaws and workarounds for IEEE 1588 (transparent) clocks". In: International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS). Oct. 2009, pp. 1–6. DOI: 10.1109/ISPCS.2009.5340204.

Treytl, Albert, Bernd Hirschler, and Thilo Sauter. "Secure tunneling of high-precision clock synchronization protocols and other time-stamped data". In: *IEEE International Workshop on Factory Communication Systems (WFCS)*. IEEE, 2010, pp. 303–312.

Tsang, Jeanette and Konstantin Beznosov. *A Security Analysis of the Precise Time Protocol*. LERSSE technical report, Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada, LERSSE-TR-2006-02, 2005.

Tsang, Jeanette and Konstantin Beznosov. "A Security Analysis of the Precise Time Protocol (Short Paper)". In: Springer, 2006, pp. 50–59.

Ueda, S., S. Kaneko, N. Kawaguchi, H. Shigeno, and K. Okada. "Authenticating Video Streams". In: *International Conference on Advanced Information Networking and Applications, (AINA)*. Apr. 2006, pp. 863–868. DOI: 10.1109/AINA.2006.107.

Ueda, Shintaro, Shin-ichiro Kaneko, Nobutaka Kawaguchi, Hiroshi Shigeno, and Ken-ichi Okada. "A Real-Time Stream Authentication Scheme for Video Streams". In: *Information and Media Technologies* 1.2 (2006), pp. 1014–1024.

Ullmann, M. and M. Vögeler. "Delay attacks - Implication on NTP and PTP time synchronization". In: *International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. Oct. 2009, pp. 1–6. DOI: 10.1109/ISPCS.2009.5340224.

Vazquez, Felix Iglesias, Robert Annessi, and Tanja Zseby. "Analytic Study of Features for the Detection of Covert Timing Channels in Network Traffic". In: *Journal of Cyber Security and Mobility* 6.3 (2017), pp. 225–270. DOI: 10.13052/jcsm2245-1439.632.

Vazquez, Felix Iglesias, Robert Annessi, and Tanja Zseby. "DAT detectors: uncovering TCP/IP covert channels by descriptive analytics". In: *Security And Communication Networks* (2016). DOI: 10.1002/sec.1531.

Vormayr, G., T. Zseby, and J. Fabini. "Botnet Communication Patterns". In: *IEEE Communications Surveys Tutorials* 19.4 (2017), pp. 2768–2796.

Wang, Qiyan, H. Khurana, Ying Huang, and K. Nahrstedt. "Time Valid One-Time Signature for Time-Critical Multicast Data Authentication". In: *IEEE INFOCOM*. Apr. 2009, pp. 1233–1241. DOI: 10.1109/INFCOM.2009.5062037.

Wegman, Mark N and J Lawrence Carter. "New hash functions and their use in authentication and set equality". In: *Journal of Computer and System Sciences* 22.3 (1981), pp. 265–279.

Wei, Dong, Yan Lu, M. Jafari, P.M. Skare, and K. Rohde. "Protecting Smart Grid Automation Systems Against Cyberattacks". In: *IEEE Transactions on Smart Grid* 2.4 (Dec. 2011), pp. 782–795. ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2159999.

Westin, Alan. *Privacy and Freedom*. Ig Publishing, 1967. ISBN: 9781935439974.

Wolf, Christopher and Bart Preneel. "Taxonomy of Public Key Schemes based on the problem of Multivariate Quadratic equations." In: *IACR Cryptology ePrint Archive* 2005 (2005), p. 77. URL: http://eprint.iacr.org/2005/077.pdf.

Wong, Chung Kei and S.S. Lam. "Digital signatures for flows and multicasts". In: *International Conference on Network Protocols*. Oct. 1998, pp. 198–209. DOI: 10.1109/ICNP.1998.723740.

Wong, Chung Kei and S.S. Lam. "Digital signatures for flows and multicasts". In: *IEEE/ACM Transactions on Networking* 7.4 (Aug. 1999), pp. 502–513. ISSN: 1063-6692. DOI: 10.1109/90.793005.

Wuille, Pieter. *BIP 0062: Dealing with Malleability - (2014) [Online]*. https://github.com/bitcoin/bips/blob/master/bip-0062.mediawiki [Accessed: 18-Sep-2018].

Yang, Qingyu, Dou An, and Wei Yu. "On time desynchronization attack against IEEE 1588 protocol in power grid systems". In: *IEEE Energytech*. May 2013, pp. 1–5. DOI: 10.1109/EnergyTech.2013.6645332.

Young, Adam and Moti Yung. "The Dark Side of "Black-Box" Cryptography or: Should We Trust Capstone?" In: *Advances in Cryptology*. Springer, 1996, pp. 89–103. ISBN: 978-3-540-68697-2. DOI: 10.1007/3-540-68697-5_8.

Zhang, Yinghui, Hui Li, Xiaoqing Li, and Hui Zhu. "Provably secure and subliminal-free variant of schnorr signature". In: *Information and communication technology-EurAsia conference*. Springer, 2013, pp. 383–391.

Zhao, Xianfeng and Ning Li. "Reversible Watermarking with Subliminal Channel". In: *International Workshop on Information Hiding*. Springer, 2008, pp. 118–131. ISBN: 978-3-540-88961-8. DOI: 10.1007/978-3-540-88961-8_9.

Zseby, T. "Is IPv6 Ready for the Smart Grid?" In: *International Conference on Cyber Security*. Dec. 2012, pp. 157–164. DOI: 10.1109/CyberSecurity.2012.27.

Zseby, Tanja and Joachim Fabini. "Security Challenges for Wide Area Monitoring in Smart Grids". In: *e & i Elektrotechnik und Informationstechnik* 131.3 (May 2014), pp. 105–111. ISSN: 0932-383X, 1613-7620. DOI: 10.1007/s00502-014-0203-3.

Zseby, Tanja, Felix Iglesias Vazquez, Valentin Bernhardt, Davor Frkat, and Robert Annessi. "A Network Steganography Lab on Detecting TCP/IP Covert Channels". In: *IEEE Transactions on Education* PP.99 (2016), pp. 1–9. DOI: 10.1109/TE.2016.2520400.