Thesis for The Degree of Licentiate of Engineering

Advanced Properties of Digital Signature Schemes

Hanna Ek

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden, 2026

"Du har det inom dig."
- IngMari Ek

Advanced Properties of Digital Signature Schemes

Hanna Ek

ISSN 1652-876X

Department of Computer Science and Engineering
Division of Computing Science
Information Security
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2026.

Advanced Properties of Digital Signature Schemes

Hanna Ek

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg

Abstract

In today's digital society it is of great importance to be able to authenticate ourselves to others and verify the identity of others without any physical interaction. Digital signatures are a well established cryptographic primitive used for authentication over the internet. They are a purely asymmetric primitive and do not require any pre-shared secrets between the authenticating and the verifying parties. However, with our ever-growing dependence on digital platforms for communication, storage and computations, *plain* digital signatures do not cover all today's needs for secure authentication.

This thesis explores advanced properties of digital signatures, particularly it contributes to the field of multi-key homomorphic signatures and threshold signatures. Multi-key homomorphic signatures allow for verification of outsourced computations to untrusted cloud services and threshold signatures distribute trust in a system and avoid a single point of failure.

Paper I examines the succinctness property of multi-key homomorphic signatures and presents the most succinct multi-key homomorphic signatures in the literature. Succinctness refers to the asymptotic size of an aggregated signature in terms of the number of individual input signatures. This is an important aspect of homomorphic signatures since the overhead of verifying the correctness of an outsourced computation depends linearly on the size of the aggregated signature.

Paper II presents an adaptively secure universally composable threshold Schnorr signature scheme from standard assumptions. Security can be proven under a variety of assumptions and in various different frameworks that capture differently strong adversaries and more or less realistic settings. Adaptive security in the universal composability framework is considered as one of the strongest security notions. Consequently, proving security in this setting provides strong guarantees for real-world deployment.

Keywords

Cryptography, Digital Signatures, Homomorphic Signatures, Multi-Key Homomorphic Signatures, Threshold Signatures, Distributed Key Generation

List of Publications

Appended publications

This thesis is based on the following publications:

[**Paper I**] D. F. Aranha, C. Boschini, **H. Ek**, E. Pagnin , *Succinct Multi Key Linearly Homomorphic Signatures for Certified Statistics Submitted, under review.*

[**Paper II**] **H. Ek**, K. Melissaris, L. Roy, *Adaptively Secure, Universally Composable Distributed Generation of Discrete-Logarithm Based Keys*
To appear in CRYPTO'26.

Acknowledgment

I want to express my sincere gratitude to my supervisor Elena Pagnin. Thank you for all the advice, guidance, and knowledge you have shared with me over the past years. I am very grateful to have you as my supervisor. I also want to thank my co-supervisor Kelsey Melissaris for the opportunity to join interesting research projects. It is a great experience working with you. A special thank you to my collaborators Cecilia Boschini, Diego F. Aranha and Lawrence Roy. I am honored to have Daniel Slamanig as my discussion leader and Dag Wedelin as my examiner for the defense of my licentiate.

I also want to thank my dear colleagues in the Crypto(Tea)m, past and current, Elena, Lucia, Adrian, Christoph, Tejas, Shai, Ivan and Kelsey. A special thank you to everyone who contributes to the PhD *fika* for making my Wednesday afternoons extra sweet. Also, thank you to all my colleagues at CSE. Although I cannot name everyone individually, please know that you are all appreciated.

Finally, I want to thank my family and friends. Mamma, the love you have given me is enough for a lifetime and I carry the memory of you with me every day. You are my biggest inspiration through your kindness, bravery, wisdom, and care of others. To my dad and sister, thanks for your never-ending encouragement and for believing in me even when I doubt myself. I would not be here today without you, and I am beyond grateful for sharing this with you. My dearest Kåre, thank you for your constant support and always cheering me on. You brighten my days. My lovely friends, I am forever grateful for all the laughter and distractions from work whenever needed.

Contents

Part I

Thesis Summary

Chapter 1

Introduction

Over the last decades, the importance of digital platforms and services has increased drastically and completely changed the way we communicate, store information, and manage our data. This gives rise to many new challenges: How do I know that I am indeed texting with my friend? How can an online bank ensure that a user is authorized to access a specific account? How can a computer verify that a software update comes from a trusted source? All these questions are fundamentally concerned with authentication.

At the core, authentication is the act of verifying the identity of *someone* or *something* and is something we do daily, for example, when we show our driving license to prove we are certified to drive, when accessing certain computer systems we answer security questions, such as "What was your pet's first name", to verify our identity, or when we log in to an online bank we use BankID[1] [1] to authenticate ourselves. In short, there are many reasons for authentication and likewise several ways of achieving it.

For the purpose of this thesis, let us first distinguish between digital and non-digital authentication. Showing your driving license is a completely non-digital authentication process, while answering a security question and authentication through BankID are both digital authentications. This leads us to yet another distinction: Cryptographic authentication and non-cryptographic authentication.

To showcase this distinction, consider an adversary who wants to forge an authentication. Such an adversary can fairly easily learn the answer to the security question by looking up common pet names or doing some lightweight investigation of an authorized person's background. Thus, within a relatively short amount of time they can authenticate themselves to a computer system to which they are not authorized. For cryptographic authentication however, we require *unforgeability* [2], which essentially means that it should be (almost) impossible to forge an identity. Consequently, a successful verification through a cryptographic authentication process implies, with (almost) 100% certainty, that someone is who or something is what they claim to be.

This thesis will focus on cryptographic authentication schemes and more

[1] BankID is Sweden's most used electronic identification system (eID)

specifically advanced properties of such schemes. In the remainder of this introduction, we will introduce some relevant concepts and then define the research questions of this thesis. We start with further separating the notion of authentication and then continue with introducing some advanced properties for cryptographic authentication schemes.

1.1 Cryptographic Authentication

Cryptographic authentication can be achieved through two cryptographic primitives; message authentication codes (MACs) and digital signatures (DS) [3], [4], where MACs are symmetric cryptographic primitives whereas DS are asymmetric. An authentication scheme is denoted symmetric if the view, in terms of known secrets and keys, of the sender and receiver is the same and asymmetric if the view differs. Therefore, the sender and receiver in a MAC scheme share a pre-established secret that is used for authentication. Remark that this pre-shared secret should be private and only known to the parties engaging in the MAC scheme.

Conversely, DS do not require a pre-established shared secret between the sender and receiver. Instead, the sender knows a secret key and the receiver, and all other parties, know the corresponding public key. This makes DS publicly verifiable and transferable. That is, anyone who knows the public key can verify a signature on a message. Additionally, verifying a digital signature also conveys information about who generated the signature, namely the party in possession of the secret key that corresponds to the public key used to verify. Finally, we remark that neither the public key, signature nor the combination of them leaks any information about the secret key.

From now on we will only consider digital signature schemes and refer to cryptographic authentication as digital signature schemes, or digital signatures in short.

1.2 Digital Signature Schemes

To get an intuition of digital signatures consider two parties, Alice and Bob, interacting as in Figure 1.1. Where Alice acts as the signer that computes and sends a signature and Bob acts as the verifier that receives and verifies the signature. Furthermore, we assume that Alice the signer knows a secret key, sk, and Bob the verifier knows the corresponding public key, pk and can associate it to Alice. To convince Bob about the authenticity of a message, m, Alice computes a signature, σ, of the message using sk. Then given the tuple (pk, σ, m) Bob can verify the signature with respect to the message. A valid signature convinces the verifier that the message m is the original, unmodified, message sent by the signer.

More formally. A digital signature scheme consists of a tuple of four probabilistic polynomial time (PPT) algorithms, $(\mathsf{setup}, \mathsf{keygen}, \mathsf{sign}, \mathsf{verify})$ [5] defined

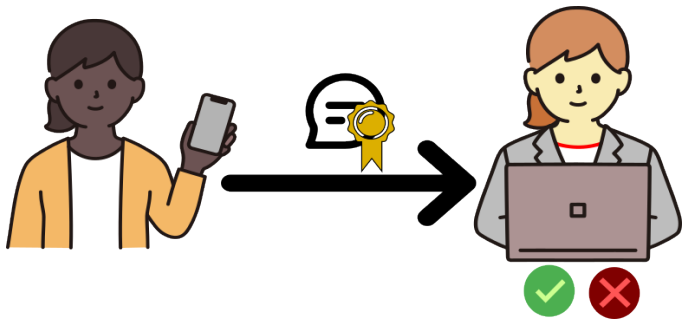

Figure 1.1: The signer (on the left) signs a message using a secret key and sends the message and the signature to the verifier (on the right). The verifier then checks the signature against the corresponding public key and either accepts or rejects the message.

below. The signer is the party that runs the keygen and sign algorithms and the verifier is the party that runs verify.

$\mathsf{setup}(\lambda, \mathsf{aux}) \to \mathsf{pp}$ **:** Takes as input the security parameter[2] λ and optionally some auxiliary input, aux, and outputs the public parameters[3] pp.

$\mathsf{keygen} \to (\mathsf{sk}, \mathsf{pk})$ **:** Outputs a random key pair, $(\mathsf{sk}, \mathsf{pk})$, where sk is the secret signing key and pk is the corresponding public verification key.

$\mathsf{sign}(\mathsf{sk}, m) \to \sigma$ **:** Takes as input a secret key, sk, and a message, m. It outputs a signature, σ.

$\mathsf{verify}(\mathsf{pk}, \sigma, m) \to 0/1$ **:** Takes as input the public key, pk, signature, σ, and a message, m. It outputs 1 if σ is deemed to be a valid signature for the message m under the public key pk and 0 otherwise.

A digital signature scheme should (at least) satisfy *correctness* and *unforgeability*.

Intuitively, correctness means that for any admissible message an honestly generated signature should be accepted by the verification algorithm with *high* probability, formally this is defined as:

$$\Pr\left[\mathsf{verify}(\mathsf{pk}, \sigma, m) = 1 \;\middle|\; \begin{array}{r} \mathsf{pp} \leftarrow \mathsf{setup}(\lambda, \mathsf{aux}) \\ (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{keygen} \\ \sigma \leftarrow \mathsf{sign}(\mathsf{sk}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).^{4}$$

Implying that for any message, m, and signature, σ, that is the output of $\mathsf{sign}(\mathsf{sk}, m)$, the verification algorithm, $\mathsf{verify}(\mathsf{pk}, \sigma, m)$ should output 0 only

[2] A security parameter determines the security level of a scheme. Typically measure in the bit length of the key, which in turn determines the algorithmic complexity of the attacks towards the scheme.

[3] The public parameters pp is an implicit input to subsequent algorithms.

[4] $\mathsf{negl}(\lambda)$ is a negligible function in the security parameter, λ. Meaning that $\mathsf{negl}(\lambda)$ is a positive function that should approach 0 faster than any inverse-polynomial in λ.

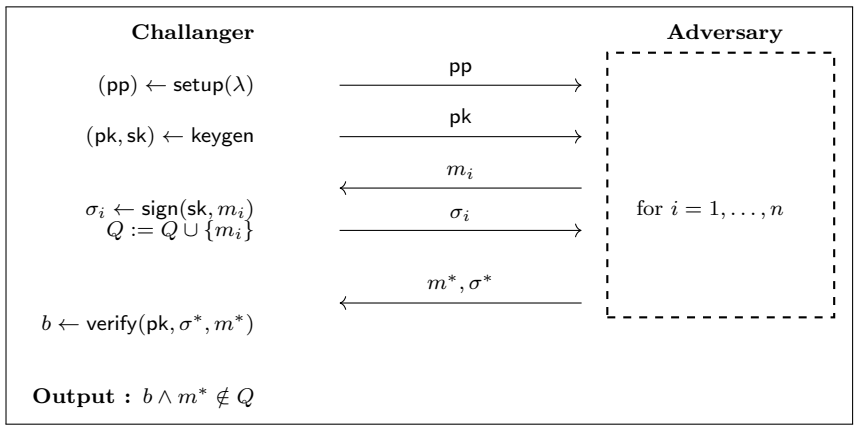

Figure 1.2: Existential unforgeability under chosen message attack, euf−cma, security game for digital signatures. The challenger samples the public parameters and a key pair, then shares the public parameters and public key with the adversary. The adversary can query for polynomially many signatures on messages of its choice and wins if it outputs a valid forgery (m^*, σ^*) such that the message m^* was not previously queried.

with negligible probability, where the probability is taken over the randomness used in the algorithms.

Unforgeability means that it should be *hard* to produce a signature, σ, of a message, m, that is accepted by the verification algorithm under some public key, pk, without access to the corresponding secret key, sk. Unforgeability is, formally, defined through the security game *existential unforgeability under chosen message attack* [2], euf−cma, shown in Figure 1.2 and a digital signature scheme, is unforgeable if:

$$\Pr\left[\mathsf{euf{-}cma}_{\mathcal{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda),$$

for all PPT adversaries, $\mathcal{A}$.

1.3 Advanced Properties of Digital Signatures

Above we described the basic structure of a digital signature scheme, in which a signer seeks to convince a verifier of the authenticity of a message. However, there exists a wide range of different digital signatures schemes for various use cases and security goals. To mention a few, we have; ring signatures [6], designated verifier signatures [7], group signatures [8], blind signatures [9], threshold signatures [10], [11] and homomorphic signatures[12], [13].

This thesis focuses on threshold signatures and homomorphic signatures, which are introduced in more detail in the following sections.

1.3.1 Homomorphic Signatures

In 2002, Johnson *et al.* [12] introduced the concept of homomorphic signatures, a way to provide authentication of computations outsourced to untrusted parties.

Consider the setting where a user Alice (acting as the signer) signs a bundle of data points $(m_1, \ldots, m_n)$ and uploads the data points and their corresponding signatures to an untrusted server. Another party, Bob (acting as the verifier) wants to evaluate a function f over the signed data. Due to limited computational power, Bob outsources the computation to the server and receives back the value m. Consequently, the following question arises: *How can the server convince Bob that the returned value m is indeed equal to the evaluation $f(m_1, \ldots, m_n)$?*

A naive approach would be to send the entire collection of messages and signatures to Bob, who can verify each signature and check that m is the intended computation. However, this would require that Bob both verifies n signatures and recomputes the entire function, which defeats the purpose of outsourcing the computation.

Homomorphic signatures address this question by enabling the server to convince Bob of such a statement without requiring Bob to recompute the function. Specifically, homomorphic signatures enables the server to compute m and derive an evaluated signature σ, from the signatures issued by the signer, that upon verification implies the correctness of the computation. In particular, verifying σ convinces the verifier that m is the evaluation of the requested function over the expected signed messages.

An important remark is that Bob does not merely want to be convinced that m is the evaluation of the function f on some data signed by Alice. Rather, he wants that m is the result of evaluating f over the exact set of data, $(m_1, \ldots, m_n)$, signed by Alice. In particular every data point in the set must be included in the computation, and no additional data may be included. To capture this homomorphic signatures are signed in a context and associated with a *label*. A label is a unique identifier of a message, for intuition it can be viewed as an address to a specific entry in a particular dataset or in a more descriptive way a timestamp of when a measurement was made.

More formally. First we introduce the notion of a *labeled programs* [13] which is defined as $\mathcal{P} := (f, \ell_1, \ldots, \ell_n)$, such that f is a function that takes n inputs and returns a single value and ℓ_i is a label associated with a message m_i. As discussed above, the notion of labels and labeled programs is essential for homomorphic signatures, as it binds the signatures to a specific context. Labels ensure that the evaluation is performed only on the intended data and prevents mix-and-match attacks in which signatures from unrelated datasets or computations are used in the evaluation to produce seemingly valid outputs.

A homomorphic signature scheme consists of the tuple of five PPT algorithms $(\mathsf{setup}, \mathsf{keygen}, \mathsf{sign}, \mathsf{eval}, \mathsf{verify})$ [13]. The algorithms $(\mathsf{setup}, \mathsf{keygen}, \mathsf{sign}, \mathsf{verify})$ are as defined in Section 1.2 with the exception that sign additionally takes as input a label, and verify takes $(\mathcal{P}, \mathsf{pk}, m, \sigma)$ as input. Thus we

refer to the definitions given above, omit them below, and merely give the definition of eval.

$\mathsf{eval}(\mathcal{P}, \mathsf{pk}, \{m_i, \sigma_i\}_{i=1}^n) \to \sigma$ Takes as input the labeled program $\mathcal{P}$, the public key pk and the messages with the corresponding signatures $\{m_i, \sigma_i\}_{i=1}^n$ and outputs an evaluated signature σ.

A homomorphic signature scheme should satisfy *unforgeability*, *authentication correctness*, *evaluation correctness* and *succinctness*. Intuitively, unforgeability, means that it should be cryptographically hard for the evaluator to produce a signature σ such that m is not the evaluation of the labeled program $\mathcal{P}$ but $\mathsf{verify}(\mathcal{P}, \mathsf{pk}, m, \sigma) = 1$. Authentication correctness means that signatures output by the signing algorithm should verify under the labeled program $\mathcal{P}_I := (I, \ell)$, where I is the identity function and ℓ is the label of the signed message. Evaluation correctness means that a signature output by eval verifies under the labeled program $\mathcal{P}$ with overwhelming probability, given the same public key and parameters. Succinctness is a fundamental property of homomorphic signatures stating that the size of the signature output by eval should be sublinear in the size of all the signatures given as input. This prevents the naive solution of defining the evaluated signatures to be the entire collection of n messages and their corresponding signatures.

Multi-Key Homomorphic Signatures. A limitation of homomorphic signatures is that they only support computations over data originating from a single signer, in contrast many practical applications which require computation over data originating from multiple independent signers. Multi-key homomorphic signature (`mkhs`) [14] schemes extend homomorphic signatures to the multi-signer setting, by enabling authenticated evaluation of functions on messages originating from multiple signers, with distinct signing keys. In a mkhs scheme eval and verify take a set of public keys as input instead of a single key.

Focusing purely on the succinctness property of multi-key homomorphic signatures, the size of the evaluated signature should be sublinear in the number of input messages and the number of participating signers. Otherwise, the trivial solution of appending all individual signatures and messages would suffice.

1.3.2 Threshold Signatures

A threshold signature [10], [11], [15] is an interactive protocol between n signers, such that any subset of at least t of the signers can collaboratively sign any admissible message m while no combination of at most $t-1$ signers can produce a valid signature. We refer to this as a (t, n)-threshold signature scheme, where $t < n$ is the threshold and n is the total number of signers in the protocol.

A core concept in threshold signatures is to split the secret key into shares that are distributed among the signers. Specifically, the key is split into n shares, one per signer in the protocol, such that any set of at most $t-1$ shares, leaks no information about the key, while any set of t shares can be used to,

for example, reconstruct the key or sign messages. Note that the secret key in its entirety is not stored in one physical location but instead distributively among the signers. This way of splitting the secret key into shares that are stored across multiple parties is a way to distribute trust and eliminate single points of failure. This increases resistance against key compromise and key-recovery attacks since an adversary must compromise at least t signers in order to recover the secret key. Notably, one of the main advantages of threshold signatures are that they enable signing of messages without needing to store the secret key in one place.

A simple and elegant approach of splitting a secret into shares that satisfies the threshold properties is through *Shamir secret sharing* [16] which is based on polynomial interpolation. Shamir secret sharing works as follows. Let $S(X)$ be a uniformly random polynomial of degree $t-1$ such that $S(0) = \mathsf{sk}$. For each party P_i, where $i \in [n]$, the share is defined as the evaluation of the polynomial at position i, $\mathtt{sh}_i = S(i)$.

This brings us to the following question: *who samples the polynomial and computes the shares?* One approach, used in for example Glacius [17] and Gargos [18], is to depend on a trusted party that samples the secret key, splits it into shares and distributes the shares among the signers. However, if one wishes to avoid relying on a trusted party, as it reintroduces a single point of failure, the key shares can instead be generated using a distributed key generation (DKG) protocol [19], [20], [21].

Distributed Key Generation. DKG is an interactive protocol between n parties, with the goal of generating a public key pk and shares $\mathsf{sh}_1, \ldots, \mathsf{sh}_n$ of the corresponding secret key sk. The protocol ensures that the public key is publicly known, while the corresponding secret key is never reconstructed or computed in its entirety. Instead, it remains distributed among the parties. At the end of the protocol each party P_i, for $i \in [n]$, obtains a threshold share, sh_i, of the secret key and learns the public key. The shares should be such that any set of at most $t-1$ shares reveals no information about the secret key, whereas any set of at least t shares can be used to, for example, jointly sign a message or reconstruct the secret key.

Note that neither threshold signatures nor DKG protocols inherently rely on Shamir secret sharing. Although Shamir secret sharing is a commonly used technique for splitting a secret into threshold shares, it is just one example of how a secret can be split into shares that achieve the desired threshold properties described above.

1.4 Proving security

Naturally, proving security properties is the most important aspect of public-key cryptography. Moreover, the most common approach for proving security of cryptographic primitives, including digital signature schemes, is through a **game-based security** definition combined with a reduction to a well-established hardness assumption, such as the discrete logarithm (DL), hardness

of factoring large numbers or learning with errors (LWE) [5].

Game-based security proofs for signature schemes result in a security argument of the form. If a polynomial time adversary can produce a valid forgery, then one can construct a polynomial time algorithms that solves a problem A. Thus, either A is solvable in polynomial time or no polynomial time adversary can produce a valid forgery and if one believes that the problem A cannot be solved in polynomial time, the scheme is unforgeable.

An example of such a security game is the $\mathsf{euf{-}cma}$ game shown in Figure 1.2. There the adversary learns the public parameters and public key of a signature scheme, and queries the challenger for signatures of arbitrary message. The goal of the adversary is to produce a valid forgery, *i.e.,* a signature of a message it has not queried. For a given signature scheme, if one proves a reduction from valid forgery to some *hard* problem, it follows that the signature scheme is unforgeable.

Game-based security proofs work well for proving security of individual primitives, but these security proofs do not necessarily extend when multiple primitives are combined into larger protocols. Thus, over the last decades, it has become common to prove security in more modular frameworks.

Universal composability [22] is a framework for proving security of cryptographic schemes that enables modular design and analysis of complex cryptographic protocols composed of simpler cryptographic building blocks. This feature makes proving security in the universal composability (UC) framework desirable.

In UC, security of a protocol is proven by showing that its execution is indistinguishable from an execution of an ideal functionality. An ideal functionality can be thought of as some utopian entity that satisfies exactly the security guarantees we want our protocol to have. The idea behind UC is then that if it is impossible for any PPT distinguisher to distinguish between these two executions, the protocol must provide the same security guarantees as the ideal functionality.

Adaptive security in the UC framework considers adversaries that may corrupt any protocol participant at any time during (or after) the execution of the protocol. Upon corruption, the adversary learns the entire state of the newly corrupted party, including all secret keys and randomness used during the execution. This state leakage upon corruption makes adaptive UC security proofs especially challenging. A protocol is adaptively UC-secure if it is secure against adaptive corruptions.

1.5 Research Questions

This thesis aims to advance the field of digital signatures. It does so by exploring two different research questions, one related to multi-key homomorphic signatures and one related to threshold signatures and DKG protocols:

Question 1: Is it possible to construct a fully succinct, multi-key homomorphic signature scheme that is practical for real-world deployment? Where fully succinct means that the size of the evaluated signature is independent of

the number of input messages to the outsourced computation and logarithmic in the number of signers contributing with signed input messages.

In 2024, Anthoine *et al.* [23] presented a multi-key homomorphic signature scheme, which evaluated signature is logarithmic in the number of input messages. Thus, if each signer contributed with only one message their results would imply the definition of fully succinct. However, in many applications, there are considerably fewer signers than messages, thus it remains an interesting open question to make the size of the evaluated signature logarithmic in the number of signers instead of the number of messages.

Question 2: Can we construct a DKG protocol that is adaptively secure in the UC framework from standard assumptions. Furthermore, can we use this construction to derive an adaptively UC secure threshold signature scheme that does not depend on trusted key distribution?

Previous adaptively secure DKG schemes use either non-standard assumptions such as secure erasure [24], [25] or weaker security models such as oracle-aided simulation [25] or the single-inconsistent player model [24], [26], [27]. Secure erasure requires honest parties to delete parts of their state that cannot be simulated upon corruption. In oracle-aided simulation the simulator is given limited access to a discrete logarithm oracle and in the single-inconsistent player model the simulator is allowed to fail if the adversary corrupts a specific random honest party.

Chapter 2

Summary of Included Papers

This thesis consists of two included papers, titled *Succinct Multi-Key Linearly Homomorphic Signatures for Certified Statistics* (Paper I) and *Adaptively Secure, Universally Composable Distributed Generation of Discrete-Logarithm Based Keys* (Paper II). Below we give a brief overview of the main contributions of the two papers and describe my specific contribution to each of them.

2.1 Paper I

I am the main technical author of this paper. I have been responsible for designing the construction as well as formulating and writing the security proofs.

This paper answers research **Question 1**, formulated in Chapter 1, positively. The paper presents an adaptive secure fully succinct multi-key *linearly*[1] homomorphic signature scheme (`mklhs`) and uses the scheme for a case-study of certified statistics over COVID-19 data.

The paper takes as starting point the efficient and simple multi-key linear homomorphic signature scheme by Aranha and Pagnin [28], whose aggregated signatures are of size $\mathcal{O}(t)$, *i.e.,* linear in the number of signers contributing with input messages to the outsourced computation and independent of n, the total number of input messages. A limitation of their construction is that it is only proven secure against weak adaptive adversaries [14], *i.e.,* adversaries that declare in advance which messages will be queried during the security game[2].

In our paper, we extend their signature scheme by combining it with an inner product argument [29], which is a proof, whose size is logarithmic in the

[1] Linear indicates that the outsourced functions are limited to linear functions.

[2] The security game capturing the security of homomorphic signatures is called homomorphic unforgeability under chosen message attack, `huf-cma`, [14].

length of the statement, of the following relation,

$$\mathcal{R} = \{(\mathbf{g}, \mathbf{h} \in \mathbb{G}^m, P \in \mathbb{G}, c \in \mathbb{Z}_q); \mathbf{a}, \mathbf{b} \in \mathbb{Z}_q^m \text{ s.t } P = \mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}} \wedge c = \langle \mathbf{a}, \mathbf{b} \rangle\}$$

As a result, we obtain aggregated signatures of size $\mathcal{O}(\log(t))$. Additionally, we prove security against adaptive adversaries which is a stronger security notion than weak-adaptive and captures a more realistic attacker scenario.

The MKHS schemes in both Paper I and [28] supports homomorphic evaluation of linear functions only, *i.e.,* the outsourced function is restricted to be a linear function, as indicated in the name `mklhs`. Besides this limitation, there are many interesting applications for our `mklhs` construction, in the paper, this is exemplified by a case study on authenticated mortality statistics of the impact of COVID-19 in Spain.

2.2 Paper II

I joined this project when a draft of the constructions had already been made. My main contribution has been to write the security proof of the DKG construction, which has been an iterative process requiring adjustments to the construction.

This paper answers research **Question 2**, formulated in Chapter 1, positively by presenting;

- A 3-round DKG protocol with Guaranteed Output Delivery (GOD). GOD of a DKG protocol states that given an honest majority the DKG protocol will output a public key pk known to all participants and a corresponding secret share sh_i to each party P_i participating in the protocol.
- Two committed[3] DKG protocols, one in 3-rounds with GOD given honest majority and one in 2-round with identifiable abort (IA). IA of a DKG protocol states that if a party is corrupt, *i.e.,* adversarial, the protocol will output the identity of this user, enabling the honest users to eliminate this party before rerunning the protocol.
- A 3-round threshold Schnorr signature with identifiable abort derived from the committed DKG protocol.

All of the above schemes are proven secure in the adaptive UC framework from standard assumption. Particularly the security relies on Decisional-Diffie-Hellman (DDH) [30], the random oracle model (ROM) [31], adaptive non-interactive zero knowledge proofs (aNIZK) [32] and a public-key infrastructure (PKI).

In summary, we present the first adaptively secure universally composable DKG and committed DKG functionalities that do not rely on non-standard assumptions such as secure erasure, single-inconsistent party or oracle aided simulation.

[3] *Committed* indicates that the protocol additionally outputs commitments to each party's secret key share, sk_i.

Chapter 3

Future Work

This chapter presents different future research directions motivated by the results in the included papers. Although both included papers aim to enhance the functionality and security of digital signature schemes, the two papers included in this thesis consider two rather different digital signatures; multi-key homomorphic signatures and threshold signatures. Consequently, the future work directions of this thesis are also split into two corresponding research directions, one building on the work with homomorphic signatures and on the work with threshold signatures. Additionally, the thesis proposes a third line of future work, considering advanced properties of post-quantum secure digital signature schemes.

3.1 Extending the Function Space for Multi-Key Homomorphic Signatures

As of today, to the best of my knowledge, all efficiently implementable `mkhs` schemes are restricted to linear functions, *i.e.,* the function f on the signed messages outsourced to the evaluator is limited to be of the form, $f(m_1, \ldots, m_n) = \sum_{i=1}^{n} a_i m_i$, where a_i are known coefficients. Indeed, this limitation also applies to the construction presented in the first included paper. However, many interesting statistical measurements, such as variance and Euclidean squared distance, are quadratic in the input messages.

Anthoine *et al.* [23] proposed two `mkhs` constructions that support boolean circuits of unbounded depth and arithmetic circuits of unbounded depth and bounded width. Their constructions rely on batch arguments (BARGs) [33], [34] and functional commitments (FC) [35] and the efficiency of their signatures scales with the underlying BARG and FC constructions. Both BARGs and FC have good asymptotic behavior but they suffer from heavy machinery and large concrete parameters making them inefficient in actual deployment. Thus, the `mkhs` constructions presented by Anthoine *et al.* are theoretically very interesting but not practical.

Consequently, an interesting line of research would be to determine if one

can construct an efficiently implementable multi-key homomorphic signature that supports quadratic computations over the input messages.

3.2 Adaptively UC Secure Threshold ECDSA Signatures

In the second paper, we introduce an adaptively UC secure committed DKG protocol and then show how to derive threshold signatures from our committed DKG protocol. More specifically, we prove threshold Schnorr Signatures *i.e.,* a threshold version of the signature scheme obtained by applying the Fiat-Shamir [36] transform to challenge-based Sigma protocols [37], [38] for the DL relation. Although, Schnorr signatures are used in mainstream applications such as Bitcoin [39], other signature schemes are more widely deployed, such as ECDSA [40] which are used in for example authentication protocols such as TLS and SSH as well as in blockchains. Therefore, a natural extension to the second included paper is to explore if the committed DKG protocols can be used to derive adaptively UC secure threshold ECDSA signatures.

3.3 Advanced Properties of Post-Quantum Secure Signature Schemes

Two famous and well studied cryptographic security assumptions are the hardness of finding the discrete logarithm of a group element and factoring large numbers. However, both of these assumptions are not post-quantum secure assumptions as both rely on the hidden periodicity problems which can be solved in polynomial time by a quantum computer [41]. Consequently, in the presence of quantum computers, all signature schemes relying on these assumptions can be forged in polynomial time. Both papers included in this thesis, rely on variations of the discrete logarithm assumption, thus their security guarantees do not hold in the post-quantum setting.

A natural line for future research is post-quantum secure digital signatures and more specifically advanced properties of post-quantum signatures. In the NIST competition *Post-Quantum Cryptography: Additional Digital Signature Schemes* [42] there are 14 signature schemes selected for the second round and 4 of them, MAYO, QR-UOV SNOVA and UOV [43], [44], [45], [46], are based on multivariate quadratic problem (MQ). Recently, advanced properties for MQ-based signature schemes has gained some attention, specifically with respect to threshold signatures, [47] [48]. Thus, an interesting line of research would be to achieve the same results as the included papers but for signature schemes based on post-quantum secure assumptions.

Bibliography

[1] Finansiell ID-Teknik BID AB, *Bankid*, https://www.bankid.com, Accessed: 2026-03-20.

[2] S. Goldwasser, S. Micali and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal on Computing*, vol. 17, no. 2, pp. 281–308, 1988. DOI: 10.1137/0217017.

[3] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976. DOI: 10.1109/TIT.1976.1055638.

[4] M. Bellare, R. Canetti and H. Krawczyk, "Keying hash functions for message authentication," in *Advances in Cryptology – CRYPTO'96*, ser. Lecture Notes in Computer Science, vol. 1109, Springer, Berlin, Heidelberg, Aug. 1996, pp. 1–15. DOI: 10.1007/3-540-68697-5_1.

[5] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, Third. Chapman and Hall, CRC Press, 2014.

[6] R. L. Rivest, A. Shamir and Y. Tauman, "How to leak a secret," in *Advances in Cryptology – ASIACRYPT 2001*, ser. Lecture Notes in Computer Science, vol. 2248, Springer, Berlin, Heidelberg, Dec. 2001, pp. 552–565. DOI: 10.1007/3-540-45682-1_32.

[7] H. Lipmaa, G. Wang and F. Bao, "Designated verifier signature schemes: Attacks, new security notions and a new construction," in *ICALP 2005: 32nd International Colloquium on Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, vol. 3580, Springer, Berlin, Heidelberg, Jul. 2005, pp. 459–471. DOI: 10.1007/11523468_38.

[8] D. Chaum and E. van Heyst, "Group signatures," in *Advances in Cryptology – EUROCRYPT'91*, ser. Lecture Notes in Computer Science, vol. 547, Springer, Berlin, Heidelberg, Apr. 1991, pp. 257–265. DOI: 10.1007/3-540-46416-6_22.

[9] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology – CRYPTO'82*, Plenum Press, New York, USA, 1982, pp. 199–203. DOI: 10.1007/978-1-4757-0602-4_18.

[10] Y. Desmedt, "Society and group oriented cryptography: A new concept," in *Advances in Cryptology – CRYPTO'87*, ser. Lecture Notes in Computer Science, vol. 293, Springer, Berlin, Heidelberg, Aug. 1988, pp. 120–127. DOI: 10.1007/3-540-48184-2_8.

[11] Y. Desmedt and Y. Frankel, "Threshold cryptosystems," in *Advances in Cryptology – CRYPTO'89*, ser. Lecture Notes in Computer Science, vol. 435, Springer, New York, Aug. 1990, pp. 307–315. DOI: 10.1007/0-387-34805-0_28.

[12] R. Johnson, D. Molnar, D. X. Song and D. Wagner, "Homomorphic signature schemes," in *Topics in Cryptology – CT-RSA 2002*, ser. Lecture Notes in Computer Science, vol. 2271, Springer, Berlin, Heidelberg, Feb. 2002, pp. 244–262. DOI: 10.1007/3-540-45760-7_17.

[13] R. Gennaro and D. Wichs, "Fully homomorphic message authenticators," in *Advances in Cryptology – ASIACRYPT 2013, Part II*, ser. Lecture Notes in Computer Science, vol. 8270, Springer, Berlin, Heidelberg, Dec. 2013, pp. 301–320. DOI: 10.1007/978-3-642-42045-0_16.

[14] D. Fiore, A. Mitrokotsa, L. Nizzardo and E. Pagnin, "Multi-key homomorphic authenticators," in *Advances in Cryptology – ASIACRYPT 2016, Part II*, ser. Lecture Notes in Computer Science, vol. 10032, Springer, Berlin, Heidelberg, Dec. 2016, pp. 499–530. DOI: 10.1007/978-3-662-53890-6_17.

[15] S. Katsumata, M. Reichle and K. Takemure, "Adaptively secure 5 round threshold signatures from MLWE/MSIS and DL with rewinding," in *Advances in Cryptology – CRYPTO 2024, Part VII*, ser. Lecture Notes in Computer Science, vol. 14926, Springer, Cham, Aug. 2024, pp. 459–491. DOI: 10.1007/978-3-031-68394-7_15.

[16] A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, no. 11, pp. 612–613, Nov. 1979. DOI: 10.1145/359168.359176.

[17] R. Bacho, S. Das, J. Loss and L. Ren, "Glacius: Threshold schnorr signatures from DDH with full adaptive security," in *Advances in Cryptology – EUROCRYPT 2025, Part II*, ser. Lecture Notes in Computer Science, vol. 15602, Springer, Cham, May 2025, pp. 304–334. DOI: 10.1007/978-3-031-91124-8_11.

[18] R. Bacho, S. Das, J. Loss and L. Ren, "Adaptively secure three-round threshold schnorr signatures from DDH," in *Advances in Cryptology – CRYPTO 2025, Part VI*, ser. Lecture Notes in Computer Science, vol. 16005, Springer, Cham, Aug. 2025, pp. 390–422. DOI: 10.1007/978-3-032-01887-8_13.

[19] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology – CRYPTO'91*, ser. Lecture Notes in Computer Science, vol. 576, Springer, Berlin, Heidelberg, Aug. 1992, pp. 129–140. DOI: 10.1007/3-540-46766-1_9.

[20] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems," in *Advances in Cryptology – EUROCRYPT'99*, ser. Lecture Notes in Computer Science, vol. 1592, Springer, Berlin, Heidelberg, May 1999, pp. 295–310. DOI: 10.1007/3-540-48910-X_21.

[21] R. Canetti, R. Gennaro, S. Goldfeder, N. Makriyannis and U. Peled, "UC non-interactive, proactive, threshold ECDSA with identifiable aborts," in *ACM CCS 2020: 27th Conference on Computer and Communications Security*, ACM Press, Nov. 2020, pp. 1769–1787. DOI: 10.1145/3372297.3423367.

[22] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *42nd Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Oct. 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888.

[23] G. Anthoine, D. Balbás and D. Fiore, "Fully-succinct multi-key homomorphic signatures from standard assumptions," in *Advances in Cryptology – CRYPTO 2024, Part III*, ser. Lecture Notes in Computer Science, vol. 14922, Springer, Cham, Aug. 2024, pp. 317–351. DOI: 10.1007/978-3-031-68382-4_10.

[24] R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Adaptive security for threshold cryptosystems," in *Advances in Cryptology – CRYPTO'99*, ser. Lecture Notes in Computer Science, vol. 1666, Springer, Berlin, Heidelberg, Aug. 1999, pp. 98–115. DOI: 10.1007/3-540-48405-1_7.

[25] H. Feng, T. Mai and Q. Tang, "Scalable and adaptively secure any-trust distributed key generation and all-hands checkpointing," in *ACM CCS 2024: 31st Conference on Computer and Communications Security*, ACM Press, Oct. 2024, pp. 2636–2650. DOI: 10.1145/3658644.3690253.

[26] S. Jarecki and A. Lysyanskaya, "Adaptively secure threshold cryptography: Introducing concurrency, removing erasures," in *Advances in Cryptology – EUROCRYPT 2000*, ser. Lecture Notes in Computer Science, vol. 1807, Springer, Berlin, Heidelberg, May 2000, pp. 221–242. DOI: 10.1007/3-540-45539-6_16.

[27] M. Abe and S. Fehr, "Adaptively secure feldman VSS and applications to universally-composable threshold cryptography," in *Advances in Cryptology – CRYPTO 2004*, ser. Lecture Notes in Computer Science, vol. 3152, Springer, Berlin, Heidelberg, Aug. 2004, pp. 317–334. DOI: 10.1007/978-3-540-28628-8_20.

[28] D. F. Aranha and E. Pagnin, "The simplest multi-key linearly homomorphic signature scheme," in *Progress in Cryptology - LATINCRYPT 2019: 6th International Conference on Cryptology and Information Security in Latin America*, ser. Lecture Notes in Computer Science, vol. 11774, Springer, Cham, Oct. 2019, pp. 280–300. DOI: 10.1007/978-3-030-30530-7_14.

[29] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, May 2018, pp. 315–334. DOI: 10.1109/SP.2018.00020.

[30] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, Florida: CRC Press, 1996.

[31] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *ACM CCS 93: 1st Conference on Computer and Communications Security*, ACM Press, Nov. 1993, pp. 62–73. DOI: 10.1145/168588.168596.

[32] A. Lysyanskaya and L. N. Rosenbloom, *Efficient and universally composable non-interactive zero-knowledge proofs of knowledge with security against adaptive corruptions*, Cryptology ePrint Archive, Report 2022/1484, 2022. [Online]. Available: https://eprint.iacr.org/2022/1484.

[33] A. R. Choudhuri, A. Jain and Z. Jin, "SNARGs for $\mathcal{P}$ from LWE," in *62nd Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Feb. 2022, pp. 68–79. DOI: 10.1109/FOCS52979.2021.00016.

[34] Y. Kalai, A. Lombardi, V. Vaikuntanathan and D. Wichs, "Boosting batch arguments and RAM delegation," in *55th Annual ACM Symposium on Theory of Computing*, ACM Press, Jun. 2023, pp. 1545–1552. DOI: 10.1145/3564246.3585200.

[35] B. Libert, S. C. Ramanna and M. Yung, "Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions," in *ICALP 2016: 43rd International Colloquium on Automata, Languages and Programming*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 55, Schloss Dagstuhl, Jul. 2016, 30:1–30:14. DOI: 10.4230/LIPIcs.ICALP.2016.30.

[36] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology – CRYPTO'86*, ser. Lecture Notes in Computer Science, vol. 263, Springer, Berlin, Heidelberg, Aug. 1987, pp. 186–194. DOI: 10.1007/3-540-47721-7_12.

[37] C.-P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, Jan. 1991. DOI: 10.1007/BF00196725.

[38] D. R. Stinson and R. Strobl, "Provably secure distributed Schnorr signatures and a (t, n) threshold scheme for implicit certificates," in *ACISP 01: 6th Australasian Conference on Information Security and Privacy*, ser. Lecture Notes in Computer Science, vol. 2119, Springer, Berlin, Heidelberg, Jul. 2001, pp. 417–434. DOI: 10.1007/3-540-47719-5_33.

[39] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, https://bitcoin.org/bitcoin.pdf, 2008.

[40] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Robust threshold DSS signatures," in *Advances in Cryptology – EUROCRYPT'96*, ser. Lecture Notes in Computer Science, vol. 1070, Springer, Berlin, Heidelberg, May 1996, pp. 354–371. DOI: 10.1007/3-540-68339-9_31.

[41] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *35th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Nov. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[42] National Institute of Standards and Technology, *Post-quantum cryptography: Additional digital signature schemes — round 2 additional signatures*, https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures, Last updated September 24, 2025, 2022. [Online]. Available: https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures.

[43] W. Beullens, F. Campos, S. Celi, B. Hess and M. J. Kannwischer, "MAYO," National Institute of Standards and Technology, Tech. Rep., 2024, available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures.

[44] R. Akiyama et al., "QR-UOV," National Institute of Standards and Technology, Tech. Rep., 2024, available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures.

[45] L. Wang et al., "SNOVA," National Institute of Standards and Technology, Tech. Rep., 2024, available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures.

[46] W. Beullens et al., "UOV — Unbalanced Oil and Vinegar," National Institute of Standards and Technology, Tech. Rep., 2024, available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures.

[47] D. Cozzo and N. P. Smart, "Sharing the LUOV: Threshold post-quantum signatures," in *17th IMA International Conference on Cryptography and Coding*, ser. Lecture Notes in Computer Science, vol. 11929, Springer, Cham, Dec. 2019, pp. 128–153. DOI: 10.1007/978-3-030-35199-1_7.

[48] S. Celi, D. Escudero and G. Niot, "Share the MAYO: Thresholdizing MAYO," in *Post-Quantum Cryptography - 16th International Workshop, PQCrypto 2025, Part I*, Springer, Cham, Apr. 2025, pp. 165–198. DOI: 10.1007/978-3-031-86599-2_6.