Karlsruhe Institute of Technology

Improvements and New Constructions of Digital Signatures

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Jessica Koch

aus Speyer

Tag der mündlichen Prüfung: 7.02.2019

Erster Referent: Prof. Dr. Dennis Hofheinz
Zweiter Referent: Prof. Dr.-Ing Tibor Jager

Danksagung

Im folgenden möchte ich mich ganz herzlich bei all denjenigen bedanken, die mich während der Anfertigung dieser Dissertation unterstützt, motiviert, begleitet und bereichert haben.

An allererster Stelle möchte ich mich bei Dennis bedanken, der diese Arbeit inhaltlich betreut hat. Ich versuche es diesmal mit einem Global-Kompliment, da meine Lokal-Komplimente leider nicht immer die erhoffte Wirkung hatten. Danke für Alles was ich von dir gelernt habe. Deine Leidenschaft für Kryptographie ist einzigartig und motivierend.

Ein besonderer Dank gilt auch Tibor, meinem Zweitgutachter, der damals während meiner Zeit als Doktorand eine hervorragende Vorlesung über digitale Signaturen gehalten hat. Dies hat mein Interesse an Signaturen stark geweckt und meine Forschung sehr beeinflusst. Dazu noch deine unverwechselbare lockere und sympathische Art, die mir in so vielem geholfen hat, danke.

Ich möchte mich auch noch ganz besonders bei Bernhard bedanken, der immer da ist, wenn man ihn braucht. Egal, ob es um Technik geht, die mich mal wieder zur Verzweiflung bringt, oder darum verrückte Sachen zu fotografieren, gemeinsam leckeren Obstsalat zu essen oder einfach nur um emotionalen Beistand.

Desweiteren gebührt auch ganz großer Dank Thomas und Lisa. Ihr habt mich beide so unglaublich unterstützt, dass ich das hier gar nicht richtig in Worte fassen kann. Lisa, auf sehr emotionale Weise und mit einer nie erschöpfenden Hilfsbereitschaft. Thomas, durch sehr viele wertvolle Gespräche und unermüdliches Korrekturlesen (ich weiß, das war bestimmt nicht immer einfach und schön für dich). Dein Formalismus ist beneidenswert, deine Hilfsbereitschaft großartig!

Danke auch Julia, die wirklich immer einen praktischen und hilfreichen Rat für mich hat, egal ob es um wissenschaftliche Themen oder Kindererziehung geht. Danke Flo, du warst unverzichtbar für das Ankommen hier, deine wunderbar sympathische Art ist Gold wert. Danke Christoph, für deine kreative Ader und der tollen Zusammenarbeit.

Danke Jörn und Willi, für die motivierenden Vorlesungen während meiner Studienzeit; Björn, für die aufbauenden Gespräche und den musikalischen Einfluss; Brandon, für die letzten witzigen Wochen vor Abgabe; Markus, für das unvergessliche und mutmachende Werwolf Spiel; Bogdan, for your beautiful pictures, your incredible BU_cake, and many more funny stuff; Geoffroy, for your legendary party stories; Jiaxin, for your help regarding signatures or strange master students; Akin, für deine schelmische Art; Matthias N., für die liebenswerte, chaotische Art, die nur Mathematiker haben können; Rebecca, für deinen liebevollen Umgang mit Nelo auf Jörns Party; Gunnar, für deine Genialität und schnellen Lösungen; Alex, für deine inspirierende Art; Andy, für deine Lockerheit; Valerie, für die witzigen Spiele; Michael, für die lustigen Erklärungen; Sven, für deinen trockenen Humor, Super Mario Kart und dein Talent Spielregeln auszunutzen; Mario, für dein ansteckendes Lachen; Anna-Louise, für das Austauschen und Mitfühlen bei den Kleinen; Holger, für das geduldige Passwort zurücksetzen bzw. ändern; Kathrin, für deine mitfühlende und verständnisvolle Art - ich wünsche dir auch viel Kraft und Durchhaltevermögen kurz vorm Ziel; Steffie, für deine lustige Art und Hilfe bei Elternzeit, Arbeitsverträgen und alles was ich sonst noch in meinem Chaos vergessen habe; Frau Manietta, für den Überblick in Allem und die ruhige, kraftspendende Aura; Dirk, für das Bild des Autoverkäufers und der Fertigen in meinem Kopf, Dr. Meta, dem Ich-möchte-ein-Spiel-spielen-Jigsaw Vergleich und vielen weiteren Lachanfällen (bitte mach deine eigene Show auf youtube!); Nico, für deine unverwechselbare, witzige direkte Art; Matthias H., für die Suppen-Gesellschaft; Matthias G., für die Hilfsbereitschaft bei Kind und Arbeit; Patrik, für die tollen Kickererlebnisse im Kap; Tobi, für die Begeisterung

beim Klettern; Antonio und Zorica, für eure Herzlichkeit und unsere Wir-Geschichten; Carmen, Christian, Daniel, David, Jeremias, Jochen, Lukas, Rafael, schön euch alle getroffen zu haben! Danke für die unvergesslichen Kickerturniere in der Mittagspause, die Doppelkopf-Runde, die tollen Spiel-und Filmabende und (sportlichen) Ausflüge.

Ich bedanke mich zudem aus tiefstem Herzen bei Stella, Leo und Uwe. Stella, für deine wunderbare Freundschaft, die mir in allen Situationen viel Kraft gegeben hat und immer geben wird. Leo, für deine außergewöhnliche Unterstützung und wertvollen Gespräche. Uwe, für unendlich viele inspirierende Gedanken und Momente.

Es gibt noch so viele Menschen, die mich auf meinem Weg unabhängig von der Arbeit bereichert oder zum Nachdenken gebracht haben und denen ich dafür danken möchte. Allen voran Carmen, Olga und Anna. Danke für die wundervolle Zeit!

Zu guter Letzt möchte ich mich noch bei meiner Familie bedanken. Meinen Eltern, vor allem meiner Mutter, die immer an mich geglaubt hat und mich so sein ließ, wie ich gewollt habe. Meinem Bruder, der mir sehr, sehr wichtig ist. Nelo und Simon, meiner eigenen kleinen Familie, die mir unendlich viel bedeutet. Danke Simon für deine Liebe und Ehrlichkeit.

Ich liebe Euch.

Contents

Zusammenfassung

Ein digitales Signaturverfahren, oft auch nur *digitale Signatur* genannt, ist ein wichtiger und nicht mehr wegzudenkender Baustein in der Kryptographie. Es stellt das digitale Äquivalent zur klassischen handschriftlichen Signatur dar und liefert darüber hinaus noch weitere wünschenswerte Eigenschaften.

Mit solch einem Verfahren kann man einen öffentlichen und einen geheimen Schlüssel erzeugen. Der geheime Schlüssel dient zur Erstellung von Signaturen zu beliebigen Nachrichten. Diese können mit Hilfe des öffentlichen Schlüssels von jedem überprüft und somit verifiziert werden.

Desweiteren fordert man, dass das Verfahren „sicher“ sein soll. Dazu gibt es in der Literatur viele verschiedene Begriffe und Definitionen, je nachdem welche konkreten Vorstellungen beziehungsweise Anwendungsgebiete man hat. Vereinfacht gesagt, sollte es für einen Angreifer ohne Kenntnis des geheimen Schlüssels nicht möglich sein eine gültige Signatur zu einer beliebigen Nachricht zu fälschen. Ein sicheres Signaturverfahren kann somit verwendet werden um die folgenden Ziele zu realisieren:

Authentizität: Jeder Empfänger kann überprüfen, ob die Nachricht von einem bestimmten Absender kommt.

Integrität der Nachricht: Jeder Empfänger kann feststellen, ob die Nachricht bei der Übertragung verändert wurde.

Nicht-Abstreitbarkeit: Der Absender kann nicht abstreiten die Signatur erstellt zu haben.

Damit ist der Einsatz von digitalen Signaturen für viele Anwendungen in der Praxis sehr wichtig. Überall da, wo es wichtig ist die Authentizität und Integrität einer Nachricht sicherzustellen, wie beim elektronischen Zahlungsverkehr, Softwareupdates oder digitalen Zertifikaten im Internet, kommen digitale Signaturen zum Einsatz.

Aber auch für die kryptographische Theorie sind digitale Signaturen ein unverzichtbares Hilfsmittel. Sie ermöglichen zum Beispiel die Konstruktion von stark sicheren Verschlüsselungsverfahren.

Eigener Beitrag. Wie bereits erwähnt gibt es unterschiedliche Sicherheitsbegriffe im Rahmen von digitalen Signaturen. Ein Standardbegriff von Sicherheit, der eine recht starke Form von Sicherheit beschreibt, wird in dieser Arbeit näher betrachtet. Die Konstruktion von Verfahren, die diese Form der Sicherheit erfüllen, ist ein vielschichtiges Forschungsthema. Dazu existieren unterschiedliche Strategien in unterschiedlichen Modellen. In dieser Arbeit konzentrieren wir uns daher auf folgende Punkte.

- Ausgehend von vergleichsweise realistischen Annahmen konstruieren wir ein stark sicheres Signaturverfahren im sogenannten Standardmodell, welches das realistischste Modell für Sicherheitsbeweise darstellt. Unser Verfahren ist das bis dahin effizienteste Verfahren in seiner Kategorie. Es erstellt sehr kurze Signaturen und verwendet kurze Schlüssel, beides unverzichtbar für die Praxis.

- Wir verbessern die Qualität eines Sicherheitsbeweises von einem verwandten Baustein, der identitätsbasierten Verschlüsselung. Dies hat unter anderem Auswirkung auf dessen Effizienz bezüglich der empfohlenen Schlüssellängen für den sicheren Einsatz in der Praxis. Da jedes identitätsbasierte Verschlüsselungsverfahren generisch in ein digitales Signaturverfahren umgewandelt werden kann ist dies auch im Kontext digitaler Signaturen interessant.

- Wir betrachten Varianten von digitalen Signaturen mit zusätzlichen Eigenschaften, sogenannte aggregierbare Signaturverfahren. Diese ermöglichen es mehrere Signaturen effizient zu einer zusammenzufassen und dabei trotzdem alle zugehörigen verschiedenen Nachrichten zu verifizieren. Wir geben eine neue Konstruktion von solch einem aggregierbaren Signaturverfahren an, bei der das Verfahren eine Liste aller korrekt signierten Nachrichten in einer aggregierten Signatur ausgibt anstatt, wie bisher üblich, nur *gültig* oder *ungültig*. Wenn eine aggregierte Signatur aus vielen Einzelsignaturen besteht wird somit das erneute Berechnen und eventuell erneute Senden hinfällig und dadurch der Aufwand erheblich reduziert.

Abstract

A digital signature scheme, or short *digital signature*, is an important and indispensable cryptographic building block. It is the digital equivalent of the classical handwritten signature and provides even more desirable features.

A digital signature scheme generates a public key and a secret key. The secret key is used to create signatures for arbitrary messages. This can be publicly verified by anyone who is aware of the public key.

Furthermore one requires that the scheme should be "secure". There are many different terms and definitions in the literature, depending on the application. Informally, it should not be possible for an adversary to forge a valid signature of an arbitrary message without knowing the secret key. A secure digital signature scheme can therefore be used to achieve the following security goals:

Authentication: Each receiver can verify whether the message comes from a specific sender.

Data Integrity: Each receiver can determine whether the message was modified during transmission.

Non-Repudiation: The sender cannot deny having created the signature.

Thus the use of digital signatures is very important for many applications in practice. Wherever it is important to ensure the authenticity and integrity of a message, such as in electronic payment transactions, software updates or digital certificates on the Internet, digital signatures are used.

Digital signatures are also a very important tool for cryptographic theory. For example, they enable the construction of strongly secure encryption schemes.

Own Contribution. As already mentioned, there are different security notions in the context of digital signatures. A standard notion of security, which describes a rather strong form of security, is examined in detail in this work. The construction of digital signature schemes that fulfill this form of security is a rich research topic. Different strategies exist in different models. In this thesis we therefore focus on the following points.

- Based on comparatively realistic assumptions, we construct a strongly secure digital signature scheme in the so-called standard model, which is the most realistic model for security proofs. Our scheme was the most efficient digital signature scheme in its category at that time. The scheme creates very short signatures and uses short keys, both essential for practical use.

- We improve the quality of a security proof of a related building block, called identity-based encryption, which allow us, e.g., to reduce the recommended key length for secure use in practice. Any identity-based encryption scheme can be generically transformed into a digital signature scheme. Thus our improvements are also interesting regarding digital signatures.

- We consider variants of digital signatures with additional features, so-called aggregate signature schemes. These allow to efficiently combine several signatures into one and still verify all associated messages. We improve verification of aggregate signatures by

constructing a new scheme, where the verification algorithm outputs a list of all correctly signed messages in an aggregate signature instead of only *valid* or *invalid*. If an aggregated signature consists of many individual signatures, re-computations and possibly re-sending of all individual signatures becomes obsolete, which reduces a lot of costs.

1 Introduction

Cryptography

Cryptography literally means "secret writing" and is originally the theory of encrypting information or messages. This science started thousands of years ago and is more important than ever nowadays. It is impossible to imagine life, especially online communication, without cryptography. Cryptography ensures that two parties can securely communicate over an insecure channel. By now, cryptography is so much more than just encrypting messages and has a wide range of applications. The research focuses for instance on constructing and designing new secure cryptographic primitives such as digital signature schemes, identity-based encryption schemes, cryptographic hash functions and many more cryptographic building blocks. It also concentrates on improving the security proofs and the efficiency of these cryptographic primitives.

Focus of this Work

In this thesis, we primarily concentrate on digital signature schemes. We are interested in constructing secure schemes and how to improve them regarding security and efficiency. We show how to instantiate strong secure digital signature schemes in chapter 3 and aggregate signature schemes which satisfy an additional property in chapter 4. The efficiency of security proofs of a related cryptographic primitive, namely identity-based encryption, is discussed in chapter 5.

But first, for a better understanding of what we mean by security and efficiency, we introduce some general terms and concepts of cryptography.

Concepts of Cryptography

In cryptography, the following terms and concepts are important for a better understanding of security and how a security proof works.

Security. Depending on the application of a cryptographic scheme, there are many different properties a scheme should fullfill, but common to all and most important is to be *secure*. It is not easy to give a general definition of security since, e.g., an encryption scheme has different security requirements than a digital signature scheme. But even for the same class of scheme, depending on the area of application and other desired features, the requirements can vary strongly.

This has led to many different security notions in the literature and we specify this in the case of digital signature schemes in more detail below. At the moment, just keep in mind that there are many notions of security and we are interested in the way we can prove a scheme secure, i.e., that we can show the scheme satisfies a specific security notion.

Reductions. The security of cryptographic schemes usually has to be based on complexity theoretic intractability assumptions, i.e., assumptions stating that it is hard to solve a given computational problem. A computational problem can be seen as an infinite collection of *instances* and is said to be *hard* if computing a solution of a random instance cannot be done in polynomial time with significant probability. Here, time is given as a function in a security parameter which is given to all parties of the system, i.e., polynomial time means that the time required is polynomial in this security parameter.

This need for assumptions is due to the fact that the mere existence of most cryptographic building blocks already implies $P \neq NP$. For instance a one-way function, which is a very basic cryptographic primitive, only exists if $P \neq NP$. Since we are unable to prove this inequality given current proof techniques, we can only make assumptions on anything beyond this. Such assumptions that are used in cryptography are derived from number theoretic problems which withstood cryptanalytic attacks for many years. A complexity theoretic intractability assumption A is said to be *stronger* than a complexity theoretic intractability assumption B, if A implies B (and the converse is false or not known). That means, even if A is false, B can still be true.

In cryptography, a *standard assumption* (or *simple assumption*) is a well-studied complexity theoretic intractability assumption, where the problem instances only depend on the security parameter and are assumed to be solved in polynomial time with at most *negligible* probability (where a negligible function converges faster to zero than the inverse of any polynomial). Since standard assumptions are comparatively weak, a security proof based on a standard assumption is preferred over a proof under a stronger, more complex and maybe even not well-studied assumption.

A *reduction* is a method for proving security of a cryptographic scheme by reducing the security of the scheme to the hardness of solving a computational problem (or to the security of an underlying cryptographic building block). In other words, if there is an algorithm (or *adversary*) that breaks the security of the respective cryptographic scheme with non-negligible probability, it can be used to construct an algorithm (or *problem solver*) that efficiently solves an instance of the computational problem. This contradicts the hardness of the computational problem and the conclusion is that there is no successful adversary on the scheme.

Standard Model. The *standard model* is a computational model in cryptography where an adversary is only limited by the amount of computation time. Security proofs in the standard model are desirable, but often difficult to achieve. Therefore, other security models were introduced, where cryptographic primitives are replaced with idealized versions. For example the best-known idealized security model is the *random oracle model* [BR93]. In the random oracle model a cryptographic hash function of the scheme is idealized as a random function, called the random oracle. One drawback of this model is that there exists artificially constructed schemes which can be proven secure in the random oracle model but are insecure with any concrete instantiation of the hash function in practice [CGH98]. Since in this work all proofs are given in the standard model, we do not give any further details. We only want to emphasize again, that a proof in the standard model is always preferred to a proof in any other model.

Cryptographic schemes which can be proven secure using only complexity theoretic intractability assumptions without using any idealization are said to be proven secure in the standard model.

Tight Security. A *tight security reduction* means, that the security of the scheme is very closely related to the hardness of a computational problem, i.e., in the security reduction the success probability and running time of the problem solver is about the same as the success probability and running time of the adversary attacking the scheme. In the most cases, this is not self-evident, since the problem solver has to "use" the adversary for his purposes and maybe has to guess, with probability $\frac{1}{L}$, for which instance of his problem the adversary is useful. This security *loss* L leads to the fact that the success probability of the problem solver is much lower than the success probability of the adversary on the scheme. In order to guarantee the same security for the scheme as the computational problem is assumed to be hard one has to adjust the parameters of the scheme, which for instance leads to longer keys. In most applications this is not desirable and a tight security reduction, where L is a constant factor, allows to choose the key length independently of, e.g., the number of expected users and is therefore preferred.

1.1 Digital Signature Schemes

A *digital signature scheme* (short: *digital signature*), roughly speaking, enables the process of signing and verifying contracts or documents in the digital world, as done in the classical sense via handwritten signatures. Digital signatures are a very powerful cryptographic tool and provide much stronger security guarantees than classical handwritten signatures.

A good introduction to this topic, which we are following here to describe digital signature schemes, is given in [Kat10] and [Jag12]. Finally, we will see why it is difficult to construct strongly secure digital signature schemes, especially in the standard model under standard assumptions.

Digital Signatures and Applications. Informally, a digital signature scheme is used to generate a *signature* for some digital data or message, usually an arbitrary bit-string, which can be verified by anyone. It consists of three algorithms, one for *key generation*, *signing* and *verification*, respectively. A *signer* generates a key pair (pk, sk), via the key generation algorithm offered by the scheme, where pk is publicly available for any *receiver* and denoted as the signer's *public key*, and sk is the signer's corresponding *secret key* or *signing key*, which is kept secret. A signature is then generated by using the signing algorithm of the scheme on input the secret key and the message. To verify a signature of a message, i.e., to check if the signature is a *valid* signature for this message, any receiver can use the verification algorithm of the scheme on input the public key together with the signature and the corresponding message. The verification algorithm outputs either 1 (i.e., *valid*) or 0 (i.e., *invalid*).

In the following we mention some important security goals that digital signatures enable to achieve.

Authentication: The receiver can verify if the signature of a message really originates from a specific sender, since a message can only be verified correctly respective a public key if the signature was created with the corresponding secret key.

Data Integrity: The receiver can be sure that the message has not been changed during transport, neither maliciously nor due to transmission errors, otherwise verification would output *invalid*.

Non-Repudiation: The sender of a signature cannot deny, at any time, having signed the corresponding message.

Note that the achievement of these security goals rely on the fact that the secret key has not been leaked or revoked. They provide important mechanisms for real-world applications. In the following we give some examples of important applications of digital signatures.

Electronic Payment Transaction: For electronic payment transaction, e.g. paying with a credit card, authenticity and integrity of messages are ensured via digital signatures. In particular, digital signatures are an important security component of the *Europay/Mastercard/VISA* framework.

Digital Certificates: A *digital certificate* is a signed electronic document, which proves the ownership of a public key. Digital certificates represent an important basic building block, for example, in a *public-key infrastructure* (PKI). A PKI is a set of roles, policies and procedures required for the creation, management, distribution, use, storage and revocation of digital certificates and the management of public-key encryption.

Transport Layer Security: The *Transport Layer Security* (TLS) *protocol* is a cryptographic protocol to ensure secure communication over a computer network. When we visit a secure website (i.e., starting with *https://*) then the web browser communicates via the TLS protocol which authenticates the communication partner via a digital certificate.

Software Updates: If we need to download an update, e.g., for our operating system from the internet the authenticity must be verified and therefore digital signatures are necessary.

Theoretical Cryptography: Digital signatures are an essential building block in crypotographic theory. They can be used to achieve stronger security goals for cryptographic schemes, e.g. public-key encryption schemes [HJ12].

History of Digital Signatures. In 1976, Diffie and Hellman [DH76] not only laid the foundations for public-key encryption, they also described the notion of a digital signature scheme for the first time. When shortly afterwards the RSA encryption algorithm was invented by Rivest, Shamir and Adleman [RSA78] it could also be used to produce primitive digital signatures although far away from secure, since it allowed meaningful homomorphic operations on signatures.

Soon afterwards, in 1979, Lamport [Lam79] developed a digital signature scheme which satisfied a relatively weak security property since it could only be used once per key pair. At first glance, this may seem unnecessary, however, it is generically possible to extend this scheme to a digital signature scheme that can be used several times. The construction is based on one-way functions, whose existence is one of the weakest assumptions in cryptography. Further constructions in the beginning were given in [MH78] and [Rab79].

In 1988, the first security requirements for digital signature schemes were defined by Goldwasser, Micali and Rivest [GMR88]. They presented a scheme, GMR signatures, which could be proven secure in a very strong sense. We describe the security notions in more detail in the next section.

It can be said that to this day a lot of progress in constructing (strongly) secure digital signature schemes has taken place and it is still important to improve. The development of new schemes which satisfy a strong security and are also efficient in terms of key and signature size, and verfication computation time is a major research topic.

Security Notions of Digital Signatures. The two main requirements we have for a digital signature scheme are *correctness* and *soundness*. Informally, correctness means "the scheme works", i.e., for each signature for any message generated via the signing algorithm of the scheme under any secret key the verification algorithm should always output *valid* on input the signature, message and the corresponding public key.

Informally, soundness means "the scheme is secure", i.e., no efficient algorithm, or adversary, breaks the respective security property of the scheme with non-negligible probability. In principle, one would like that it is not possible to forge a valid signature of a message without knowing the secret key.

[GMR88] first introduced several strong *security notions*. A security notion is formalized as the combination of an *attack model*, which describes the ability and power of an adversary, and an *attack result*, which describes the goal an adversary has to achieve to break the security of the scheme. There are many security notions depending on the model and the goals, but we only focus on two important notions regarding the relevance for this work. For formal definitions see chapter 2.

Existentially unforgeable under non-adaptive chosen-message attack (EUF-naCMA): Before an adversary sees the public key of the scheme he has to choose a list of messages. Then the adversary gets the public key and corresponding signatures for the chosen messages. The goal of the adversary is to forge a valid signature of a new arbitrary message of his choice.

Existentially unforgeable under chosen-message attack (EUF-CMA): The adversary first gets the public key and can then *adaptively* choose messages and obtains the corresponding signatures, where adaptively means depending on the public key or previous signatures. The goal of the adversary is again to forge a valid signature of a new arbitrary message of his choice.

For the most real-world applications, a digital signature scheme, which satisfies the EUF-CMA-notion, or in other words which is EUF-CMA-*secure*, is preferred. EUF-CMA security is a very strong security notion and has become the standard security notion for digital signature schemes. EUF-CMA is stronger than EUF-naCMA, i.e., EUF-CMA security implies EUF-naCMA security.

Security Experiment. In order to prove security formally, one has to define the desired security notions precisely. One possibility to achieve a precise description is via a security experiment. A *security experiment* describes a "game" between two parties, the *challenger* and the adversary. Intuitively, the adversary *wins* the game, if he breaks the security of the scheme. For example, on a high level, we give the EUF-CMA security experiment in the following:

1. The challenger generates a public and secret key of the scheme and forwards the public key to the adversary.
2. The adversary chooses a message and sends it to the challenger. The challenger signs the message using the secret key and sends the signature back. The adversary can adaptively repeat this polynomially often.
3. Finally, the adversary outputs a *forgery*, consisting of a message and a corresponding signature.

The adversary wins the game, if the forgery signature is valid, i.e., the verification algorithm outputs *valid* on input the public key, the forgery message and the corresponding signature, and if the forgery message was never queried to the challenger, i.e., the adversary did not see a valid signature for the forgery message before.

To prove security of a given scheme, the security will be, as already mentioned, *reduced* to some complexity theoretic intractability assumption, i.e., to the task of solving a hard computational problem. A problem solver is playing the security experiment game with an adversary attacking the scheme. The problem solver gets a computational challenge and has to simulate the security experiment for the adversary. Therefore, the problem solver embeds his challenge in such a clever way that he can use a successful adversary for his purposes. This means, if the adversary outputs a valid forgery with non-negligible probability, the problem solver can use this to solve his challenge also with non-negligible probability. This contradicts the hardness of the problem and, hence, such an adversary must have a negligible probability of success.

Problems in Constructing Strongly Secure Digital Signatures. We think that an essential problem in constructing strongly secure digital signature schemes is that in the security reduction the adversary wins if he outputs a valid signature for any message of his choice, which was not queried to the problem solver. This could correspond to the choice of many different problem instances for the problem solver. Therefore, the problem solver has to guess correctly for which instance the adversary will forge a signature, to be able to solve his own challenge. Since there are superpolynomially many possible messages for the adversary to choose and just as many problem instances for the problem solver, this results in a negligible probability of solving the computational problem even if the adversary has a high probability of success.

Several reduction strategies which overcome this problem and enable to prove digital signature schemes EUF-CMA-secure are known in the literature, e.g. [Cor00; Wat05; HK12a; Boy10; CD96]. However, most of the schemes are not proven secure in the standard model [Cor00] or are proven secure under non-standard assumptions [CS00; BB08]. Many of them also require large public keys or generate large signatures, both of which are not desirable.

1.1.1 Contribution to Digital Signatures

In [BHJKS13; BHJKSS13] we introduce a new strategy to overcome this problem, dubbed *confined guessing*, and give EUF-CMA-secure instantiations under different standard assumptions.

We emphasize here that this technique and the generic construction are not part of this thesis and are discussed in more detail in the thesis "*On Cryptographic Building Blocks and Transformations*" [Str15].

Contribution. In this work, the main focus lies on the instantiation of EUF-CMA-secure digital signature schemes, which we obtain by applying the confined guessing technique. In particular, we concentrate on an instantiation proven EUF-CMA-secure under the *Computational Diffie-Hellman* (CDH) assumption in the standard model. Informally, the CDH assumption states that it is intractable in certain cyclic groups to compute a group element g^{ab} only given the group elements g, g^a and g^b. This assumption is a well-studied, relatively weak assumption.

We first instantiate a digital signature scheme which only satisfies a very mild form of security under the CDH assumption, which usually can be achieved easier than full, i.e. strong, security. Then, applying a generic transformation, we show an efficient construction of an EUF-CMA-secure CDH-based digital signature scheme which uses $\log(k)$ (for a security parameter k) instances of the mildly secure digital signature scheme, where we are now in a position to reduce the EUF-CMA security of the new scheme to the mild security of the underlying scheme.

Thus, a signature of the new scheme consists of $\log(k)$ signatures of the underlying scheme. This can be optimized to constant size by *aggregation* (further explained in the next section). However, we need additional elements in the public key, which results in $\log(k)$ elements.

We achieve an EUF-CMA-secure CDH-based digital signature scheme proven secure in the standard model with constant size signatures and logarithmic size public key. At that time, this was the first strongly secure digital signature scheme with such short signatures and public key proven secure in the standard model under a very simple assumption.

Possible Optimizations and Further Progress. One way to improve a digital signature scheme, already strongly secure under a standard assumption in the standard model, is to improve the efficiency of the scheme. This can be done either in reducing key and signature size or in terms of a tight security reduction.

As already mentioned and applied to optimize our scheme, aggregation is a possibility to reduce the size of signatures or the number of signatures that need to be transmitted. Therefore, we will take a closer look on how this is done and what aggregate signature schemes are in general in the next section.

Regarding tight security reductions we will focus in this work on a variant of public-key encryption schemes, denoted as identity-based encryption schemes, which are closely related to digital signature schemes. As we will see, each identity-based encryption scheme can be generically transformed into a digital signature scheme and thus, a tight security reduction in the context of identity-based encryption is also interesting for digital signatures. We give a brief description in section 1.3.

1.2 Aggregate Signature Schemes

An aggregate signature scheme is an extension of an ordinary digital signature scheme, which allows to generate an *aggregate signature* (short: *aggregate*), that compresses many single signatures of different users on distinct messages.

Aggregate Signatures and Applications. Suppose there are l signers with l different public keys, who want to sign l different messages for the same receiver with a conventional digital signature scheme. Then the receiver has to verify all of these message-signature pairs, where the total bit-length of these pairs grows linearly in the number of signers. If they want to reduce this overhead concerning storage space, bandwidth and computation time and cost, they can use an aggregate signature scheme instead. In an aggregate signature scheme, additionally to the algorithms of a digital signature scheme, there is a public aggregation algorithm. This aggregation algorithm takes as input a set of l signatures, corresponding to l public keys and

messages, and outputs an aggregate signature, which can be used to verify all of these signatures at once. The aggregate signature is of approximately the same size as a single individual signature of the scheme.

In 2003, the first aggregate signature scheme was proposed by Boneh, Gentry, Lynn and Shacham (BGLS) [BGLS03] in the random oracle model. From then on, a lot of progress happened and many aggregate signature schemes were constructed. Most of them, like BGLS, using bilinear maps [GR06; BNN07; RS09; HKW15]. The first construction of such a scheme in the standard model using multilinear maps was given in [HSW13]. Since it has been proven difficult to construct full aggregate schemes in the standard model, a lot of research was focused on signature schemes with some form of restricted aggregation. Most proposals outside the bilinear (or multilinear) setting require the signers to aggregate in a "sequential way" (denoted as *sequential aggregate signature schemes*), where each signer has to add his individual signature to the aggregate one after the other by using his own secret key [LMRS04a; Nev08; FLS12; LLY13]. That means no public aggregation is possible here anymore.

The applications of aggregate signature schemes are numerous and can be found everywhere one wants to save bandwith and storage space or needs faster verification. For example, in a public-key infrastructure of *depth* l, each user is given a chain of l digital certificates. The chain contains l signatures on l different certificates, each issued by l different Certificate Authorities. Using an aggregate signature scheme the size can be reduced. Another application is in the field of *sensor networks*, where each sensor measures specific data of the environment and sends this data to a base station, where the data has to be verified or is collected to be sent to another station. Since bandwith and computational power of the sensors and base stations are limited aggregate signatures are preferred. The list of possible applications can be continued by *authenticating software*, *secure logging*, *secure routing* and so on.

Problem. However, one problem which arises in all applications is the fact that the verification algorithm only outputs *valid* or *invalid*, independently of which signature or how many signatures in the aggregate are valid or invalid. In other words, if just one faulty signature is contained in the aggregate the verification algorithm outputs *invalid*, one can not make any conclusions about the remaining correctly signed signatures and the whole aggregate is useless even if the majority is still correctly signed. This leads to re-computing and re-sending all affected data which should be verified via the invalid aggregate. For example in sensor networks, all measured data is lost if one sensor sends an invalid signature, e.g. because of transmission errors, to the base station for further aggregation. Usually the base station has not enough computational power to ensure the validity of each signature before adding them to an aggregate, since verification often requires expensive computations.

1.2.1 Contribution to Aggregate Signatures

In [HKKKR16] we addressed this problem and developed the concept of *fault-tolerant aggregate signature schemes*.

Contribution. In this work, we present the first instantiation of a fault-tolerant aggregate signature scheme. Such an aggregate signature scheme can tolerate up to a specific number d of invalid or faulty signatures contained in one aggregate. The verification algorithm of this scheme does not only output *valid* and *invalid* but instead a list which contains all correctly signed messages and omits the invalidly signed messages.

We achieve this by using a *d-cover-free family* first introduced by Kautz and Singleton [KS64]. A d-cover-free family is a combinatorial structure and related to error-correcting codes. Informally, such a family can be represented as a matrix M where the entries are only 1 or 0 and which is able to "handle" up to d errors.

This matrix gives us a rule how we have to aggregate in our new scheme. Each column corresponds to an individual signature. Each 1 entry is replaced with the corresponding signature.

Aggregation is done row-wise, i.e., each signature in a row is aggregated, using an underlying ordinary aggregate signature scheme. This results in a vector τ of aggregated signatures, which is the aggregate signature of our fault-tolerant aggregate signature scheme. The size of the vector corresponds to the number of rows of the matrix, which is not of constant-size as in the most ordinary aggregate signature schemes. The same signature is aggregated several times in different rows and this leads to some kind of redundancy in our vector. A valid ordinary aggregate signature in the vector is sufficient to verify the validity of all contained signatures. If we have less than d invalid individual signatures in our aggregate signature, due to the structure of the d-cover free family, it is still possible by verifying each component of the vector separately to determine all valid signatures (for more details see chapter 4).

This generic construction can be done with any underlying aggregate signature scheme and any d-cover-free family. In this work we present a concrete instantiation of a fault-tolerant aggregate signature scheme using a d-cover-free family based on polynomials over a finite field [KRS99].

1.3 Identity-Based Encryption Schemes

Identity-based encryption (IBE) is a very important cryptographic primitive. It is related to public-key encryption (PKE), where a sender can securely communicate with a receiver by only knowing the receiver's public key. In a PKE scheme the sender uses the public key to encrypt a message and the receiver uses a secret key corresponding to the public key for decryption. To ensure that a public key is authentic, it must be signed by a trusted third party. The introduction of additional infrastructure to certify the authenticity of public keys and to revoke keys lead to a difficult key-management problem. Furthermore, it is not guaranteed that public keys are not compromised.

Identity-Based Encryption Schemes. The idea of identity-based encryption (IBE) is to solve or at least to simplify these problems and it was first proposed by Shamir [Sha84] in 1984. In an IBE scheme, in contrast to ordinary PKE schemes, an arbitrary identifier of the receiver (such as an e-mail address or IP address) and a set of global public parameters are sufficient to encrypt a message. This eliminates the need to distribute a separate public key for each user in the system. An authenticated user obtains a corresponding (user) secret key for decrypting from a trusted authority, which only needs the user's identifier and a master secret key.

In 2001, the first (efficient) realizations were published by Boneh and Franklin [BF01] and Cocks [Coc01], both secure in the random oracle model. Since that time, a great progress was made in achieving IBE schemes that are secure in the standard model [CHK07; BB04a; BB04b; Wat05; Wat09]. In [Wat09] a new proof strategy, denoted as dual system encryption (see below for a brief description and chapter 5 for more details), was introduced, which leads to very efficient IBE schemes based on standard assumptions and satisfying a very strong notion of security.

Security Notion for IBE Schemes. Informally, the standard notion for security in public-key encryption schemes demands that it is infeasible for an adversary, i.e., only possible with negligible probability, given a ciphertext encrypted under a known public key to learn anything about the corresponding message except for the length. In particular, the adversary is allowed to choose two arbitrary messages of the same length and receives one of them encrypted as a challenge ciphertext. The scheme is secure in this sense, if no adversary can efficiently determine which message was encrypted with a probability significantly better than just simple guessing. Regarding identity-based encryption the challenge ciphertext is encrypted under a chosen identity and additionally the adversary is allowed to see revealed user secret keys for other identities of his choice.

For a long time it was not easy to prove that an IBE scheme satisfies this notion of full adaptively security in the standard model and even under standard assumption.

Dual System Encryption Proof Strategy. As already mentioned, in 2009 Waters [Wat09] introduced *dual system encryption*, a novel and powerful proof technique, to construct full adaptively secure IBE schemes in the standard model under standard assumptions.

In a dual system setting, ciphertexts and user secret keys can take on two types: normal or semi-functional. Semi-functional ciphertexts and user secret keys are not used in the real system, they are only needed in the security proof. A normal user secret key can decrypt normal or semi-functional ciphertexts, and a normal ciphertext can be decrypted by normal or semi-functional user secret keys, for the same identity, respectively. However, when a semi-functional key is used to decrypt a semi-functional ciphertext for the same identity, decryption will fail.

Security for dual systems is proved, as often in security proofs, using a sequence of games which are shown to be indistinguishable, i.e., if an adversary could distinguish them he can be used to solve a hard computational problem. The first game is the real security game (with normal ciphertext and user secret keys). In the next game, the challenge ciphertext is semi-functional, while all the user secret keys are normal. Then a series of games follow in which step by step one user secret key after the other is changed to semi-functional. In the last game all the user secret keys and the challenge ciphertext given to the adversary are semi-functional. Hence, none of the given user secret keys can be useful for decrypting the challenge ciphertext. At this point, it is possible to replace the ciphertext by a random message and the adversary can not do better than guessing. Thus, proving security becomes relatively easy.

1.3.1 Contribution to Identity-Based Encryption

In [HKS15] we introduce a new variant of the dual system proof technique, dubbed *extended nested dual system groups*, to be able to prove almost tight and full adaptive security of an IBE scheme in the multi-instance, multi-ciphertext setting. The multi-instance, multi-ciphertext setting models a more realistic scenario, which is desirable. In this scenario we have many instances of an IBE scheme and in each instance the adversary obtains many challenge ciphertexts to distinguish for different identities.

Contribution. In this work, we present the first instantiation of extended nested dual system groups (ENDSG) in composite-order pairing-friendly groups. ENDSGs are based on nested dual system groups (NDSG), developed by Chen and Wee [CW13], which are based on the dual system framework introduced by Waters [Wat09].

NDSGs enable to prove the first IBE scheme almost tightly secure in the standard model under standard assumptions in the single-instance, single-ciphertext setting. The dual system proof strategy usually requires, as already mentioned, a game based approach, where each game hop is reduced to a computational problem. This results in a security loss in size of the number of games. In [Wat09], the number of games is approximately the same as the number of user secret keys revealed to the adversary. In [CW13], the number of games is approximately the same as the number of bits of the challenge identity, which only depends on the security parameter and is independent of the number of user secret keys. This is considered as *almost tight*.

ENDSGs enable to prove the first IBE scheme almost tightly secure in the standard model under standard assumptions in the multi-instance, multi-ciphertext setting.

Relations between IBE Schemes and Signature Schemes. An observation by Naor mentioned in [BF01] describes the relation between IBE schemes and digital signature schemes. It states, that any IBE scheme can be generically transformed in a digital signature scheme. The exact transformation is given in chapter 5. This means, that any progress regarding IBE schemes, e.g. achieving tighter security proofs or shorter parameters, can be transferred to the field of digital signature schemes. This is also interesting in view of our result to achieve almost tight security proofs for digital signature schemes in the multi-user scenario.

Own Publications

Conference papers

- Florian Böhl and Dennis Hofheinz and Tibor Jager and Jessica Koch and Christoph Striecks. Confined Guessing: New Signatures From Standard Assumptions. *Journal of Cryptology*, 28(1):176-208, January 2015.
- Dennis Hofheinz and Jessica Koch and Christoph Striecks. Identity-Based Encryption with (Almost) Tight Security in the Multi-instance, Multi-ciphertext Setting. In *Jonathan Katz, editor, Public-Key Cryptography – PKC 2015*, pages 799–822. Springer, March 2015.
- Gunnar Hartung and Björn Kaidel and Alexander Koch and Jessica Koch and Andy Rupp. Fault-Tolerant Aggregate Signatures. In *Public-Key Cryptography–PKC 2016*, pages 331-356. Springer Berlin Heidelberg, 2016.
- Gunnar Hartung and Björn Kaidel and Alexander Koch and Jessica Koch and Dominik Hartmann. Practical and Robust Secure Logging from Fault-Tolerant Sequential Aggregate Signatures. In *International Conference on Provable Security*, pages 87-106, Springer, Cham, 2017.

2 Preliminaries

Notation. For $n \in \mathbb{R}$, let $[n] := \{1, \ldots, \lfloor n \rfloor\}$. Throughout this thesis, $k \in \mathbb{N}$ denotes the security parameter. We assume, if not stated explicitly, it is implicitly given to all algorithms in the unary representation 1^k. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$.

For a probabilistic algorithm A, we write $y \leftarrow A(x)$ for the process of running A on input x with uniformly chosen random coins, and assigning y the result. To make the random coins r explicit, we write $A(x; r)$. If A's running time is polynomial in k, then A is called *probabilistic polynomial-time* (PPT).

For two random variables X, Y, we denote with $\mathsf{SD}(X\,;\,Y)$ the *statistical distance* of X and Y. We might also say that X and Y are ε-close if $\mathsf{SD}(X\,;\,Y) \leq \varepsilon$.

Definition 1. A function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if

$$\forall c \in \mathbb{R}\ \exists k_0 \in \mathbb{N}\ \forall k \geq k_0 : f(k) \leq 1/k^c,$$

i.e., it vanishes faster than the inverse of any polynomial.
On the other hand, f is *significant* if

$$\exists c \in \mathbb{R}, k_0 \in \mathbb{N}\ \forall k \geq k_0 : f(k) \geq 1/k^c,$$

i.e., it dominates the inverse of some polynomial.

Definition 2 (Bilinear map)**.** Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be cyclic groups of order N. A *bilinear map* $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ has the following properties:

Bilinearity. For all $g_1, g_1' \in \mathbb{G}_1$ and $g_2, g_2' \in \mathbb{G}_2$ it holds

$$e(g_1 g_1', g_2) = e(g_1, g_2)e(g_1', g_2) \quad \text{and} \quad e(g_1, g_2 g_2') = e(g_1, g_2)e(g_1, g_2')$$

This implies $e(g_1^a, g_2) = e(g_1, g_2^a) = e(g_1, g_2)^a$, for $a \in \mathbb{Z}_N$.

Non-Degeneracy. For generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$:

$$e(g_1, g_2) \text{ is a generator of } \mathbb{G}_T.$$

If $N = p \in \mathbb{P}$ this is equivalent to $e(g_1, g_2) \neq 1$.

Efficiently computable. The map e is efficiently computable.

If $\mathbb{G}_1 = \mathbb{G}_2$ we say e is a *symmetric* map, otherwise *asymmetric*.

Definition 3 (Group generation algorithm)**.** A *group generation algorithm* is a PPT algorithm Grp as follows:

Group Generation. $\mathsf{Grp}(1^k, n)$, on input 1^k and an integer $n \in \mathbb{N} \setminus \{0\}$, outputs a tuple of the form

$$(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, g, h, p_1, \ldots, p_n, g_{p_1}, \ldots, g_{p_n}, h_{p_1}, \ldots, h_{p_n}, e).$$

$\mathbb{G}, \mathbb{H}$ and $\mathbb{G}_T$ are descriptions of groups of order

$$|\mathbb{G}| = |\mathbb{H}| = |\mathbb{G}_T| = N = p_1 \cdot \ldots \cdot p_n,$$

for k-bit primes $p_1, \ldots, p_n \in \mathbb{P}$, with generators g and h, respectively, and $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ is a bilinear map. $g_{p_1}, \ldots, g_{p_n}$ and $h_{p_1}, \ldots, h_{p_n}$ are generators of the (proper) subgroups $\mathbb{G}_{p_i} \subset \mathbb{G}$ and $\mathbb{H}_{p_i} \subset \mathbb{H}$ of order

$$|\mathbb{G}_{p_i}| = |\mathbb{H}_{p_i}| = p_i,$$

for $i \in \{1, \ldots, n\}$, respectively.

Especially, if $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a symmetric bilinear map, we omit the group $\mathbb{H}$ and the elements h, h_{p_i}, for $i \in \{1, \ldots, n\}$ and write

$$(\mathbb{G}, \mathbb{G}_T, N, g, g_{p_1}, \ldots, g_{p_n}, e) \leftarrow \mathsf{Grp}(1^k, n),$$

unless it is necessary for a better understanding.
If $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ and $n = 1$, we omit the integer n and write

$$(\mathbb{G}, g, \mathbb{G}_T, p, e) \leftarrow \mathsf{Grp}(1^k).$$

Throughout this thesis, we only write the output which we need and omit the rest. For instance if the prime order of the subroups are secret, and if we are only interested in the group $\mathbb{G}$, we write

$$(\mathbb{G}, g, N, g_{p_1}, \ldots, g_{p_n}) \leftarrow \mathsf{Grp}(1^k, n) \text{ or}$$
$$(\mathbb{G}, g, p) \leftarrow \mathsf{Grp}(1^k)$$

Digital Signature Schemes

In the following, we give a formal definition of a digital signature scheme, which is the basis of most of the following constructions in the next chapters.

Definition 4 (Signature scheme)**.** A signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ with message space $\mathcal{M}_k$ consists of three PPT algorithms:

Key Generation. The setup algorithm $\mathsf{Gen}(1^k)$, on input the security parameter 1^k, outputs a public key pk and a secret key sk.

Signing. The signing algorithm $\mathsf{Sig}(sk, M)$, on input the secret key sk and a message $M \in \mathcal{M}_k$, outputs a signature σ.

Verification. The verification algorithm $\mathsf{Ver}(pk, M, \sigma)$, on input the public key pk, a message M, and a signature σ, outputs a bit $b \in \{0, 1\}$. Intuitively, the case $b = 1$ corresponds to a valid signature on the message, and the case $b = 0$ corresponds to invalid.

We require Σ to be *correct* in the sense that for any $k \in \mathbb{N}$, all $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, all $M \in \mathcal{M}_k$, and all $\sigma \leftarrow \mathsf{Sig}(sk, M)$, the output of $\mathsf{Ver}(pk, M, \sigma) = 1$.

Security Notions for Digital Signature Schemes. Informally, the desired security guarantee a signature scheme should offer is that no *efficient* ((probabilistic) polynomial time) adversary is able to "forge" a valid message/signature pair with respect to an honestly generated public key pk. There are many different notions regarding the abilities and interactions of the adversary (or forger F) and which security goals are sufficient to break. We concentrate on a very strong notion, called *existential unforgeable under chosen-message attacks* [GMR88], which is desirable to achieve. We also give a formal definition of a weaker, but related notion, called *existential unforgeable under non-adaptive chosen-message attacks*, where each signature scheme satisfying this notion can be transformed to a signature scheme that satisfies the strong notion (see also chapter 3, Lemma 5). Both notions are satisfied, if an adversary has only negligible advantage in winning the following corresponding security experiments, formalized in Definition 5, Definition 6 and Figure 2.1 and briefly described below:

Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}nacma}}_{\Sigma,F}(\mathrm{k})$	**Experiment** $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\Sigma,F}(\mathrm{k})$
$(M_i)_{i\in[q]} \leftarrow F(k)$	$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$	$(M^*, \sigma^*) \leftarrow F^{\mathsf{Sig}(sk,\cdot)}(pk)$
$\sigma_i \leftarrow \mathsf{Sig}(sk, M_i)$, for all $i \in [q]$	**if** $\mathsf{Ver}(pk, M^*, \sigma^*) = 1$
$(M^*, \sigma^*) \leftarrow F(state, pk, \sigma_1, \ldots, \sigma_q)$	**and** F has not queried $\mathsf{Sig}(sk, M^*)$
if $\mathsf{Ver}(pk, M^*, \sigma^*) = 1$	**return** 1
and $M^* \notin \{M_i\}_{i\in[q]}$	**else**
return 1	**return** 0
else	
return 0	

Figure 2.1: EUF-naCMA and EUF-CMA experiment for signature schemes.

Definition 5 (Existential unforgeability under non-adaptive chosen-message attacks)**.** We say a signature scheme Σ is *existentially unforgeable under non-adaptive chosen-message attacks* (EUF-naCMA-secure) if

$$\mathsf{Adv}^{\mathsf{euf\text{-}nacma}}_{\Sigma,F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}nacma}}_{\Sigma,F}(k) = 1\right],$$

is negligible for any PPT adversary F, where $\mathsf{Exp}^{\mathsf{euf\text{-}nacma}}_{\Sigma,F}(k)$ is defined in Figure 2.1 and briefly described below.

Description of $\mathsf{Exp}^{\mathsf{euf\text{-}nacma}}_{\Sigma,F}$:

- The adversary F outputs a set of messages $(M_i)_{i\in[q]}$, for which F wants corresponding signatures (non-adaptively), where $q = q(k)$ is the total number of signature queries.
- The experiment generates a key pair $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$, computes signatures $\sigma_i \leftarrow \mathsf{Sig}(sk, M_i)$, for all $i \in [q]$, and provides F with $(pk, \sigma_1, \ldots, \sigma_q)$.
- Finally, F outputs a forgery (M^*, σ^*).

The adversary F *wins* the experiment iff $\mathsf{Ver}(pk, M^*, \sigma^*) = 1$ and $M^* \notin \{M_1, \ldots, M_q\}$.

Definition 6 (Existential unforgeability under chosen-message attacks)**.** We say a signature scheme is *existentially unforgeable under chosen-message attacks* (EUF-CMA-secure) if

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\Sigma,F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\Sigma,F}(k) = 1\right]$$

is negligible for any PPT adversary F, where $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\Sigma,F}(k)$ is defined in Figure 2.1 and briefly described below.

Description of $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\Sigma,F}$:

- The experiment generates a key pair $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ and provides F with pk.
- During the experiment F has access to a $\mathsf{Sig}(sk, \cdot)$-oracle, to adaptiveley query signatures for messages $M \in \mathcal{M}_k$ under pk.
- Finally, F outputs a forgery (M^*, σ^*).

The adversary F *wins* the experiment iff $\mathsf{Ver}(pk, M^*, \sigma^*) = 1$ and F has not queried $\mathsf{Sig}(sk, M^*)$.

Standard Assumptions

In the following, we give two examples of well known standard assumptions in cryptography, which are relevant for this work, to construct useful cryptographic building blocks and for introducing related assumptions in the next chapters.

The Discrete Logarithm (DL) Problem. For a concrete instantiation of a chameleon hash function defined in chapter 3, we require the discrete logarithm problem.

Definition 7 (DL assumption)**.** We say that the *discrete logarithm (DL) assumption* holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$ if

$$\mathsf{Adv}^{\mathsf{dl}}_{\mathsf{Grp},A}(k) := \Pr\left[A(1^k, g, g^x) = x\right]$$

is negligible for any PPT adversary A, where $(\mathbb{G}, g, p) \leftarrow \mathsf{Grp}(1^k)$ and $x \leftarrow \mathbb{Z}_p$ is uniformly chosen.

The Decisional Diffie-Hellman (DDH) Problem. Since some of our standard assumptions in the following chapters are based on the decisional Diffie-Hellman Problem or related to it, we give a formal definition.

Definition 8 (DDH assumption)**.** We say that the *Decisional Diffie-Hellman (DDH) assumption* holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$ if

$$\mathsf{Adv}^{\mathsf{ddh}}_{\mathsf{Grp},A}(k) := |\Pr\left[A(1^k, g, g^a, g^b, g^{ab}) = 1\right] - \Pr\left[A(1^k, g, g^a, g^b, g^c) = 1\right]|$$

is negligible for any PPT adversary A, where $(\mathbb{G}, g, p) \leftarrow \mathsf{Grp}(1^k)$ and $a, b, c \leftarrow \mathbb{Z}_p$ are uniformly chosen.

3 A Strongly Secure Digital Signature Scheme

The design of efficient digital signature schemes that can be proven strongly secure under reliable standard assumptions in the standard model is one of the most important goals in cryptographic research.

Digital signature schemes can be built from any one-way function [Lam79; NY89; Rom90]. However, this generic construction is not particularly efficient. For instance, each signature contains $\mathbf{O}(k^2)$ preimages. As so often in real life the proved secure schemes with the best security guarantees are not nearly as efficient as the signature schemes that are used in practice, where applications require fast signing and verification along with short public keys and short signatures. One could hope that for *concrete* assumptions (such as the RSA or Diffie-Hellman-related assumptions), it is possible to derive much more efficient schemes.

Most efficient digital signature schemes follow a hash-and-sign paradigm (rather than a tree-based approach [CS00; HW09b; Fis02]) to obtain efficient schemes, in particular with short public keys and signatures. However, each of the hash-and-sign signature schemes requires relatively strong assumptions, e.g., random oracles [BLS04; GJKW07; GPV08], strong RSA assumption [CS00; CS00; GHR99] or other very specific strong assumptions [BB04c; Wat09; HK12b; CL04].

In this chapter we construct a rather efficient, stateless and compact EUF-CMA-secure signature scheme based on the Computational Diffie-Hellman (CDH) assumption. To achieve this, we use a new technique, dubbed *confined guessing*, which was published in [BHJKS13; BHJKSS13]. Most of the content of the following chapter is taken from these works, partly verbatim.

As far as we know, there were only two signature schemes based on the standard CDH assumption in the standard model at that time [HW09a; Wat05]. In [HW09a] a stateful signature scheme was constructed, i.e., the signer needs to maintain certain states. This property is not desirable in general but is acceptable in some applications. However, they achieve constant size public keys and signatures.

Therefore, [Wat05] was the only construction for stateless signature schemes that has been proven secure under the standard CDH assumption in the standard model so far. However, the signature scheme has constant size signatures but linear size public keys.

Our Contribution

Our construction is based on the signature scheme of [HW09a], but we remove the state and apply the technique of confined guessing. Thus, we are able to construct a rather efficient and EUF-CMA-secure signature scheme under a standard assumption. In our CDH-based signature scheme public keys and signatures contain $\mathbf{O}(\log(k))$ and $\mathbf{O}(1)$ group elements, respectively. At that time, this scheme was the first fully secure and stateless CDH-based signature scheme with such short public keys and short signatures proven secure in the standard model.

3.1 Organization

In section 3.2 we introduce some useful cryptographic primitives and the standard CDH assumption on which the security of our scheme is based.

In section 3.3 we introduce tag-based signatures and the concept of confined guessing. In contrast to an ordinary signature scheme, a tag-based signature scheme signs a message along

Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}dnacma}}_{\Sigma,F}(\mathrm{k})$
$(M_i)_{i\in[q]} \leftarrow F(1^k)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
$\sigma_i \leftarrow \mathsf{Sig}(sk, M_i)$, for all $i \in [q]$
$(M^*, \sigma^*) \leftarrow F(state, pk, (\sigma_i)_{i\in[q]})$
if $\mathsf{Ver}(pk, M^*, \sigma^*) = 1$
  and $\forall i \neq j : M_i \neq M_j$
  and $M^* \notin \{M_i\}_{i\in[q]}$
  return 1,
else
  return 0

Figure 3.1: EUF-dnaCMA experiment for signature schemes.

with a tag t. Confined guessing is a generic transformation from a tag-based signature scheme which satisfies a relatively mild form of security to an ordinary signature scheme which satisfies EUF-CMA security.

In section 3.4 we give an instantiation of our tag-based scheme based on the CDH assumption. We apply the transformation and some optimizations to achieve an instantiation of a rather efficient fully secure CDH-based signature scheme.

It is also possible to construct fully secure signature schemes under other standard assumptions as briefly mentioned in section 3.5.1 for RSA and the Short Integer Solution (SIS) assumption. We emphasize here that the generic transformation, and the instantiations based on RSA and SIS, are not part of this thesis. We only state the relevant results and parts which we need for our CDH-based instantiation and explain the approach in a nutshell for a better understanding in section 3.3. For more details on this, we refer to [BHJKS13; BHJKSS13] and the thesis *"On Cryptographic Building Blocks and Transformations"* [Str15].

3.2 Preliminaries

In this section we give some definitions and notations which are necessary for this chapter.

Definition 9 (Distinct-message existential unforgeability under (non-adaptive) chosen-message attacks)**.** We say a signature scheme is *existential unforgeable under distinct-message non-adaptive chosen-message attacks* (EUF-dnaCMA-secure) if

$$\mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\Sigma,F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}dnacma}}_{\Sigma,F}(k) = 1\right],$$

is negligible for any PPT adversary F, where $\mathsf{Exp}^{\mathsf{euf\text{-}dnacma}}_{\Sigma_t,F}(k)$, is defined in Figure 3.1.

The Computational Diffie-Hellman (CDH) Problem. For a concrete instantiation of the confined guessing technique (see section 3.4) we require the Computational Diffie-Hellman (CDH) assumption.

Definition 10 (CDH assumption)**.** We say that the *Computational Diffie-Hellman (CDH) assumption* holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$ if

$$\mathsf{Adv}^{\mathsf{cdh}}_{\mathsf{Grp},A}(k) := \Pr\left[A(1^k, g, g^a, g^b) = g^{ab}\right]$$

is negligible for any PPT adversary A, where $(\mathbb{G}, g, \mathbb{G}_T, p, e) \leftarrow \mathsf{Grp}(1^k)$ and $a, b \leftarrow \mathbb{Z}_p$ are uniformly chosen.

Pseudorandom Functions. Pseudorandom functions are a very useful cryptographic primitive. Informally, a pseudorandom function is a family of efficiently computable functions which are indistinguishable from a truly random function. They are a very useful tool to construct other secure cryptographic primitives.

Definition 11 (Pseudorandom Function)**.** For any set $\mathcal{S}$ a *pseudorandom function (PRF) with range* $\mathcal{S}$ is an efficiently computable function $\mathsf{PRF}^{\mathcal{S}} : \{0,1\}^k \times \{0,1\}^* \to \mathcal{S}$. We may also write $\mathsf{PRF}^{\mathcal{S}}_\kappa(x)$ for $\mathsf{PRF}^{\mathcal{S}}(\kappa, x)$ with key $\kappa \in \{0,1\}^k$. Additionally we require that for any PPT algorithm A

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF}^{\mathcal{S}},A}(k) := \left|\Pr\left[A^{\mathsf{PRF}^{\mathcal{S}}_\kappa(\cdot)}(1^k) = 1 \text{ for } \kappa \leftarrow \{0,1\}^k\right] - \Pr\left[A^{\mathcal{U}_{\mathcal{S}}(\cdot)}(1^k) = 1\right]\right|$$

is negligible in k where $\mathcal{U}_{\mathcal{S}}$ is a truly uniform function to $\mathcal{S}$.

Note that for any efficiently samplable set $\mathcal{S}$ with uniform sampling algorithm Samp we can generically construct a PRF with range $\mathcal{S}$ from a PRF $\mathsf{PRF}^{\{0,1\}^k}$ by using the output of $\mathsf{PRF}^{\{0,1\}^k}_\kappa$ as random coins for Samp. Following this principle we can construct $(\mathsf{PRF}^{\mathcal{S}_i})_{i\in[n]}$ for a family of sets $(\mathcal{S}_i)_{i\in[n]}$ from a single PRF $\mathsf{PRF}^{\{0,1\}^k}$ with sufficiently long output (hence we need only one key κ).

Chameleon Hashing. A *hash function* is a function that compresses data of arbitrary size, i.e., of size $\{0,1\}^*$, to a bit-string of fixed size, e.g. $\{0,1\}^k$, the *hash value*, depending on the security parameter k. A chameleon hash function is a very helpful cryptographic tool. Informally, a chameleon hash function is a collision-resistant hash function, although collisions can be easily computed by knowing a corresponding trapdoor.

Chameleon hash functions were first formalized by Krawczyk and Rabin [KR00]. One important application of chameleon hash functions is a generic transformation of an EUF-naCMA-secure signature scheme to an EUF-CMA-secure signature scheme ([HW09c], see also Lemma 5 in section 3.3).

Definition 12 (Chameleon Hash Scheme)**.** A *chameleon hash scheme* consists of two PPT algorithms (CHGen, $\mathsf{CHTrapColl}$).

Generation. $\mathsf{CHGen}(1^k)$, on input a security parameter 1^k, outputs a tuple (CH, τ) where CH is the description of an efficiently computable *chameleon hash function* $\mathsf{CH} : \mathcal{M} \times \mathcal{R} \to \mathcal{N}$ which maps a message M and randomness r to a hash value $\mathsf{CH}(M, r)$.

Collision. $\mathsf{CHTrapColl}(\tau, M, r, M')$, on input a trapdoor τ, arbitrary M, r, M', computes r' with $\mathsf{CH}(M, r) = \mathsf{CH}(M', r')$, and outputs r'.

We require that the distribution of r' is uniform given only CH and M'

We require collision-resistance in the sense that it is infeasible to find $(M, r) \neq (M', r')$ with $\mathsf{CH}(M, r) = \mathsf{CH}(M', r')$ without knowing the trapdoor τ.

Definition 13 (Collison-Resistance)**.** For any PPT algorithm A, the function

$$\mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},A}(k) := \Pr[A(\mathsf{CH}) = (M, r, M', r')],$$

such that $\mathsf{CH}(M, r) = \mathsf{CH}(M', r')$ with $(M, r) \neq (M', r')$, is negligible in k, for $(\mathsf{CH}, \tau) \leftarrow \mathsf{CHGen}(1^k)$.

Construction. Krawczyk and Rabin also provided a discrete-logarithm-based instantiation, which is interesting for our CDH-based instantiation in section 3.4. The construction is as follows:

Let $(\mathbb{G}, g, p) \leftarrow \mathsf{Grp}(1^k)$. Let $(\mathsf{CHGen}, \mathsf{CHTrapColl})$ be the chameleon hash function

$$\mathsf{CH} : \ \mathbb{Z}_p \times \mathbb{Z}_p \to \mathbb{G},$$

where

Generation. $\mathsf{CHGen}(1^k)$, on input a security parameter 1^k, chooses $x \leftarrow \mathbb{Z}_p$, computes $h := g^x$, and outputs a tuple

$$(\mathsf{CH}, \tau) := ((g, h), x),$$

such that, given $(M, r) \in \mathbb{Z}_p \times \mathbb{Z}_p$, the chameleon hash value is computed as

$$\mathsf{CH}(M, r) = g^M h^r.$$

Collision. $\mathsf{CHTrapColl}(\tau, M, r, M')$, on input a trapdoor $\tau = x$, arbitrary M, r, M', computes

$$r' \equiv \frac{M - M'}{x} \bmod p.$$

such that

$$g^M h^r = g^{M'} h^{r'}$$

and outputs r'.

Theorem 1. For any PPT adversary A, given $\mathsf{CH} = (g, h)$ and computing $(M, r, M', r') \in \mathbb{Z}_p^4$ with success probability ε_A, such that $(M, r) \neq (M', r')$ and

$$g^M h^r = g^{M'} h^{r'},$$

there exists a PPT adversary B, who solves the discrete logarithm problem in $\mathbb{G}$ with success probability $\varepsilon_B \geq \varepsilon_A$.

The proof can be found in [KR00].

3.3 Generic Transformation: From Mild to Full Security

We now briefly describe the confined guessing technique, a generic transformation from mildly secure tag-based signature schemes to fully secure signature schemes (without tags). The following definitions, theorems and explanations in this section are entirely adopted from [BHJKS13; BHJKSS13] and a further discussion on this topic can be found in [Str15]. Let us first define the notion of tag-based signature schemes.

Definition 14. A tag-based signature scheme $\Sigma_t = (\mathsf{Gen_t}, \mathsf{Sig_t}, \mathsf{Ver_t})$ with message space $\mathcal{M}_k$ and tag space $\mathcal{T}_k$ consists of three PPT algorithms.

Key generation. $\mathsf{Gen_t}(1^k)$ takes as input a security parameter and outputs a key pair (pk, sk).

Sign. $\mathsf{Sig_t}(sk, M, t)$, on input a secret key sk, message M, and tag t, computes a signature σ.

Verification $\mathsf{Ver_t}(pk, M, \sigma, t)$, on input a public key pk, message M, signature σ, and a tag t, outputs a bit $b \in \{0, 1\}$.

For *correctness*, we require for any $k \in \mathbb{N}$, all $(pk, sk) \leftarrow \mathsf{Gen_t}(1^k)$, all $M \in \mathcal{M}_k$, all $t \in \mathcal{T}_k$, and all $\sigma \leftarrow \mathsf{Sig_t}(sk, M, t)$ that $\mathsf{Ver_t}(pk, M, \sigma, t) = 1$.

We define a mild security notion for tag-based signature schemes, dubbed EUF-dnaCMA$^*_m$ security, which requires an adversary F to initially specify all (distinct) messages M_i it wants signed, along with corresponding tags t_i. After that, F gets to see a public key, and is subsequently expected to produce a forgery σ^* for an arbitrary fresh message M^*, but with respect to an already used tag $t^* \in \{t_i\}_i$. As a slightly technical (but crucial) requirement, we only allow F to initially specify at most m messages M_i with tag $t_i = t^*$. We call m the *tag-collision parameter*; it influences key and signature sizes, and the security reduction.

Definition 15 (EUF-dnaCMA$^*_m$)**.** Let $m \in \mathbb{N}$. A tag-based signature scheme Σ_t is *existentially unforgeable under distinct-message non-adaptive chosen-message attacks with m-fold tag-collisions* (short: EUF-dnaCMA$^*_m$ secure) if

$$\mathsf{Adv}^{\mathsf{euf\text{-}dnacma}^*_\mathsf{m}}_{\Sigma_t,F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}dnacma}^*_\mathsf{m}}_{\Sigma_t,F}(k) = 1\right]$$

is negligible for any PPT adversary F. Here, experiment $\mathsf{Exp}^{\mathsf{euf\text{-}dnacma}^*_\mathsf{m}}_{\Sigma_t,F}(k)$ is defined in Figure 3.2.

Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}dnacma}^*_\mathsf{m}}_{\Sigma_t,F}(k)$
$(M_j, t_j)_{j\in[q]} \leftarrow F(1^k)$
$(pk, sk) \leftarrow \mathsf{Gen_t}(1^k)$
$\sigma_j \leftarrow \mathsf{Sig_t}(sk, M_j, t_j)$ for all $j \in [q]$
$(M^*, \sigma^*, t^*) \leftarrow F(state, pk, (\sigma_j)_{j\in[q]})$
if $\mathsf{Ver_t}(pk, M^*, \sigma^*, t^*) = 1$
  and $\forall i \neq j : M_i \neq M_j$
  and $M^* \notin \{M_j\}_{j\in[q]}$
  and $|\{j : t_j = t^*\}| \leq m$
  and $t^* \in \{t_j\}_{j\in[q]}$
  return 1
else
  return 0

Figure 3.2: EUF-dnaCMA$^*_m$ experiment for tag-based signature schemes.

In this section, we recapitulate how to use an EUF-dnaCMA$^*_m$ secure scheme Σ_t to build an EUF-dnaCMA secure scheme Σ. This transformation is not a part of this thesis, but is necessary for the following section. (Full EUF-CMA security can then be achieved using a chameleon hash function [KR00]. We will see this explicit in a concrete instantiation in the next section.)

Description of the Tag Space. The signature scheme Σ constructed below (see also Figure 3.3) assigns to each message M a vector of tags $(t_1, \ldots, t_l)$, where each tag is derived from the message M by applying a pseudorandom function as $t_i := \mathsf{PRF}^{\mathcal{T}_i}_\kappa(M)$. A Σ-signature is of the form $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_l)$, where each $\sigma_i \leftarrow \mathsf{Sig_t}(sk, M, t_i)$ is a signature according to Σ_t with message M and tag t_i.

To this end, we separate the tag space $\mathcal{T}_k$ into $l := \lfloor \log_c(k) \rfloor$ *pairwise disjoint* sets $\mathcal{T}_i$, such that $|\mathcal{T}_i| = 2^{\lceil c^i \rceil}$. Here $c > 1$ is a *granularity parameter* that will affect key and signature sizes, and the security reduction, and is specified globally. For instance, if $c = 2$ and $\mathcal{T}_k = \{0,1\}^k$, then we may set $\mathcal{T}_i := \{0,1\}^{2^i}$.

The crucial idea is to define the sets $\mathcal{T}_i$ of allowed tags as sets quickly growing in i. This means that $(m+1)$-tag-collisions (i.e., the same tag t_i being chosen for $m+1$ different signed messages) are very likely for small i, but become quickly less likely for larger i.

Signature Scheme Σ. Concretely, let $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a signature scheme and let $\Sigma_t = (\mathsf{Gen_t}, \mathsf{Sig_t}, \mathsf{Ver_t})$ be a tag-based signature scheme with message space $\mathcal{M}_k$ and tag space $\mathcal{T}_k = \{0,1\}^k = \bigcup_{i=1}^{l} \mathcal{T}_i$, let $m \in \mathbb{N}$ and $c > 1$, and let PRF be a PRF with range $\mathcal{T}_k$.

Key generation. $\mathsf{Gen}(1^k)$, on input 1^k, runs $(pk', sk') \leftarrow \mathsf{Gen_t}(1^k)$, chooses a uniformly random PRF key $\kappa \leftarrow \{0,1\}^k$, and outputs a key pair $(pk, sk) := ((pk', \kappa), (pk, sk'))$.

Sign. $\mathsf{Sig}(sk, M)$, on input a secret key sk and a message M, computes tags $t_i := \mathsf{PRF}^{\mathcal{T}_i}_\kappa(M)$ and signatures $\sigma_i \leftarrow \mathsf{Sig_t}(sk', M, t_i)$ for $i \in [l]$, and outputs a signature $\sigma := (\sigma_i)_{i=1}^l$.

$\mathsf{Gen}(1^k)$	$\mathsf{Sig}(sk, M)$	$\mathsf{Ver}(pk, M, \sigma = (\sigma_i)_{i=1}^l)$
$(pk', sk') \leftarrow \mathsf{Gen_t}(1^k)$ $\kappa \leftarrow \{0,1\}^k$ $pk := (pk', \kappa)$ $sk := (pk, sk')$ **return** (pk, sk)	$t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$ for $i \in [l]$ $\sigma_i \leftarrow \mathsf{Sig_t}(sk', M, t_i)$ **return** $\sigma := (\sigma_i)_{i=1}^l$	$t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$ for $i \in [l]$ **return** $\bigwedge_{i=1}^{l} \mathsf{Ver_t}(pk', M, \sigma_i, t_i)$

Figure 3.3: The EUF-dnaCMA secure signature scheme.

Verification $\mathsf{Ver}(pk, M, \sigma)$, on input a public key pk, a message M and a purported signature $\sigma = (\sigma_i)_{i=1}^l$, deterministically computes tags $t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M)$ for $i \in [l]$, and outputs $\bigwedge_{i=1}^{l} \mathsf{Ver_t}(pk', M, \sigma_i, t_i)$.

Figure 3.3 provides an overview over these algorithms.
It is straightforward to verify Σ's *correctness*: If Σ_t is correct, then Σ is correct.

Theorem 2. If PRF is a PRF and Σ_t is an EUF-dnaCMA$^*_m$ secure tag-based signature scheme, then Σ is an EUF-dnaCMA-secure signature scheme. Concretely, let F be an EUF-dnaCMA forger on Σ that makes $q = q(k)$ signature queries and has advantage $\varepsilon := \mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\Sigma,F}(k)$ with $\varepsilon > 1/p(k)$ for infinitely many $k \in \mathbb{N}$. Then there exists an EUF-dnaCMA$^*_m$ forger F' on Σ_t that makes $q'(k) \leq 2 \cdot \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)}\right)^{c/m} + l \cdot q$ non-adaptive signature queries and has advantage $\varepsilon' := \mathsf{Adv}^{\mathsf{euf\text{-}dnacma^*_m}}_{\Sigma_t,F'}(k)$, and a PRF distinguisher with advantage $\varepsilon_{\mathsf{PRF}}$ such that

$$\varepsilon' \geq \frac{\varepsilon}{2} - \varepsilon_{\mathsf{PRF}} - \frac{p'(k)}{|\mathcal{M}_k|}$$

for infinitely many k, where $p'(k)$ is a suitable polynomial, and $\mathcal{M}_k$ denotes Σ_t's (and thus Σ's) message space.

The proof can be found in [BHJKS13; BHJKSS13].
We restate the idea of the reduction strategy in the following and mention some useful lemmata and requirements which are necessary in the next section.

Idea of the Reduction Strategy. We first give an intuition why Σ is EUF-dnaCMA secure. For this purpose, we map an adversary F on Σ's EUF-dnaCMA security to an adversary F' on Σ_t's EUF-dnaCMA$^*_m$ security. Intuitively, F' will internally simulate the EUF-dnaCMA security experiment for F and embed its own Σ_t-instance (with public key pk') in the Σ-instance of F by setting $pk := pk'$. Additionally, the seed κ for PRF is chosen internally by F'.

Say that F makes $q = q(k)$ (non-adaptive) signing requests for messages $M_1, \ldots, M_q$. To answer these q requests, F' can obtain signatures under pk' from its own EUF-dnaCMA$^*_m$ experiment. The corresponding tags are chosen as in $\Sigma.\mathsf{Sig}$, as $t_i^{(j)} = \mathsf{PRF}_\kappa^{\mathcal{T}_i}(M_j)$, $j \in [q]$. Once F produces a forgery $\sigma^* = (\sigma_i^*)_{i=1}^l$ for a message M^*, F' will try to use $\sigma_{i^*}^*$ (with tag $t_{i^*}^* = \mathsf{PRF}_\kappa^{\mathcal{T}_{i^*}}(M^*)$ for some appropriate $i^* \in [l]$) as its own forgery.

Indeed, $\sigma_{i^*}^*$ will be a valid Σ_t-forgery (in the EUF-dnaCMA$^*_m$ experiment) if

(a) F' did not initially request signatures for more than m messages for the forgery tag $t_{i^*}^*$

(b) $t_{i^*}^*$ already appears in at least one of F''s initial signature requests.

Our technical handle to make this event likely is a suitable choice of i^*. First, recall that the i-th signature σ_i uses $\lceil c^i \rceil$-bit tags. We will hence choose i^* such that

(i) the probability of an $(m+1)$-tag-collision among the $t_{i^*}^{(j)}$, $j \in [q]$, is significantly lower than F's success probability (so F will sometimes have to forge signatures when no $(m+1)$-tag collision occurs), and

(ii) $|\mathcal{T}_{i^*}| = 2^{\lceil c^{i^*} \rceil}$ is polynomially small (so all tags in $\mathcal{T}_{i^*}$ can be initially queried by F').

The following Lemma 3 is helpful for the analysis of selecting the challenge index in Lemma 4.

Lemma 3 ([HJK11b], Lemma 2.3)**.** Let A be a set with $|A| = a$. Let $X_1, \ldots, X_q$ be q independent random variables, taking uniformly random values from A. Then the probability that there exist $m+1$ pairwise distinct indices $i_1, \ldots, i_{m+1}$ such that $X_{i_1} = \cdots = X_{i_{m+1}}$ is upper bounded by $\frac{q^{m+1}}{a^m}$.

The proof can be found in [HJK11b].

Lemma 4. For index

$$i^* := \left\lceil \log_c \left(\log_2 \left(\frac{2 \cdot q^{m+1}}{\varepsilon(k)} \right)^{1/m} \right) \right\rceil \tag{3.1}$$

it holds that

$$\Pr\left[\exists \text{ distinct } j_0, \ldots, j_m \in [q] \text{ with } t_{i^*}^{(j_0)} = \cdots = t_{i^*}^{(j_m)}\right] \leq \frac{\varepsilon(k)}{2}, \tag{3.2}$$

and

$$|\mathcal{T}_{i^*}| \leq p''(k) \tag{3.3}$$

for a suitable polynomial $p''(k)$ and all $k \in \mathbb{N}$ with $\varepsilon(k) \geq 1/p(k)$.

Proof. From Lemma 3, we obtain

$$\Pr\left[\exists \text{ distinct } j_0, \ldots, j_m \text{ with } t_{i^*}^{(j_0)} = \cdots = t_{i^*}^{(j_m)}\right] \leq \frac{q^{m+1}}{|\mathcal{T}_{i^*}|^m} = \frac{q^{m+1}}{2^{m \cdot \lceil c^{i^*} \rceil}} \overset{(3.1)}{\leq} \frac{q^{m+1}}{\left(\frac{2q^{m+1}}{\varepsilon(k)}\right)} = \frac{\varepsilon(k)}{2}.$$

Furthermore,

$$|\mathcal{T}_{i^*}| = 2^{\lceil c^{i^*} \rceil} \overset{(3.1)}{\leq} 2 \cdot 2^{c \cdot \log_2((2q^{m+1}/\varepsilon(k))^{1/m})} = 2 \cdot \left(\frac{2q^{m+1}}{\varepsilon(k)}\right)^{c/m} \overset{\varepsilon(k) \geq 1/p(k)}{\leq} 2 \cdot \left(2p(k)q^{m+1}\right)^{c/m},$$

which is bounded by a suitable polynomial $p''(k)$. □

Now, in a next step, we are able to use known techniques to convert an EUF-dnaCMA secure signature scheme into an EUF-CMA secure signature scheme, i.e., by applying a chameleon hash function, as already used in previous works like in [HW09c].

Lemma 5 ([HW09c], Lemma 2.3)**.** Assuming the signature scheme Σ is EUF-naCMA secure and let $\mathsf{CH} : \mathcal{M} \times \mathcal{R} \to \mathcal{N}$ be a chameleon hash function, there exists a signature scheme Σ' which is EUF-CMA secure. Concretely,

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\Sigma', F}(k) - \frac{1}{|\mathcal{N}|} \;\leq\; \mathsf{Adv}^{\mathsf{euf\text{-}nacma}}_{\Sigma, F'}(k),$$

where F and F' are PPT forgers in the EUF-CMA and the EUF-naCMA security experiment, respectively.

The proof can be found in [HW09c].
Thus, we can apply Lemma 5 to transform our EUF-dnaCMA secure signature scheme from Theorem 2 into an EUF-CMA secure signature scheme.

Corollary 1. Applying Lemma 5 to the EUF-dnaCMA secure signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ yields an EUF-CMA secure signature scheme $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$.

The proof is conceptual similar to the proof of Lemma 5 and we refer to [BHJKS13; BHJKSS13] and [Str15] for further details.

3.4 Instantiation: A CDH-Based Scheme

The confined guessing technique enables instantiations of very efficient fully secure digital signatures schemes. In this section we present a concrete CDH-based instantiation. This is the main result in this thesis related to the confined guessing concept.

Following the described technique as before, we construct a full-fledged EUF-CMA-secure signature scheme based on the CDH assumption. We start by constructing a tag-based scheme under the CDH assumption, which achieves EUF-dnaCMA$^*_m$ security. Our construction is derived from the stateful CDH-based scheme of Hohenberger and Waters [HW09a], and we prove it EUF-dnaCMA$^*_m$-secure.

In particular, in the security experiment, the adversary is non-adaptive, i.e. chooses distinct messages along with not necessarily distinct tags in advance before seeing the public key. The winning condition requires the adversary to re-use a previously used tag for his forgery (M^*, t^*, σ^*). In other words, one tag has to be recycled. Thus, the adversary has to commit to a part of his forgery (M^*, t^*, σ^*) before he even sees the public key. This restriction can be exploited to embed a CDH-problem in the public key associated with t^* in a way that enables to extract a solution of that problem using the forgery.

Then we can apply the generic transformation described in section 3.3 to achieve full EUF-CMA security. Furthermore, we illustrate some optimizations that allow us to reduce the size of signatures, for instance using aggregation. Finally, we obtain compact public keys, which only require $\mathbf{O}(\log(k))$ group elements, and short signatures of constant size $\mathbf{O}(1)$. At that time, our scheme was the first fully secure and stateless CDH-based signature scheme with such compact public keys and short signatures proven secure in the standard model.

3.4.1 Construction

Parameters. From now on, we consider finite groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, generated by $(\mathbb{G}, g, \mathbb{G}_T, p, e) \leftarrow \mathsf{Grp}(1^k)$, where $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map, (i.e., $e(g, g) \neq 1$ for $g \neq 1$ and $e(g^a, g^b) = e(g, g)^{ab}$ for $a, b \in \mathbb{Z}$).

Our message space in this construction is $\mathcal{M} = \mathbb{Z}_p$ and for arbitrary messages $M \in \{0,1\}^*$, we consider an appropriate collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$. (Technically, if $\mathbb{G}$ is not fixed for a given security parameter, then a fixed message space can be, e.g., $\mathbb{Z}_{2^\ell}$, where 2^ℓ lower bounds all possible $p = |\mathbb{G}|$ for this security parameter.)

Our tag space in this construction is $\mathcal{T} = \{0,1\}^k$ and thus a tag t is a k-bit string, which can be interpreted as an element of $\mathbb{Z}_p$ since $p > 2^k$.

Tag-Based Signature Scheme Σ_t^{CDH}. We construct our CDH-tag-based signature scheme $\Sigma_t^{\mathsf{CDH}} = (\mathsf{Gen}_\mathsf{t}, \mathsf{Sig}_\mathsf{t}, \mathsf{Ver}_\mathsf{t})$ with message space $\mathcal{M} = \mathbb{Z}_p$ and tag space $\mathcal{T} = \{0,1\}^k$ as follows (see also Figure 3.4):

Key Generation. $\mathsf{Gen}_\mathsf{t}(1^k)$, on input 1^k, runs $(\mathbb{G}, g, \mathbb{G}_T, p, e) \leftarrow \mathsf{Grp}(1^k)$, samples a uniform random exponent $a \leftarrow \mathbb{Z}_p$, random group elements $u_0, \ldots, u_m, z, h \leftarrow \mathbb{G}$ and outputs $(pk, sk) := ((\mathbb{G}, g, \mathbb{G}_T, p, e, g^a, u_0, \ldots, u_m, z, h), (pk, a))$.

Sign. $\mathsf{Sig}_\mathsf{t}(sk, M, t)$, on input sk, message $M \in \mathbb{Z}_p$ and tag $t \in \{0,1\}^k$, samples $s \leftarrow \mathbb{Z}_p$, computes $\mathbf{u}_M := u_0 \prod_{i=1}^{m} u_i^{M^i}$, and outputs $(\tilde{\sigma}_1, \tilde{\sigma}_2) := ((\mathbf{u}_M)^a (z^t h)^s, g^s)$.

$\mathsf{Gen}_{\mathsf{t}}(1^k)$	$\mathsf{Sig}_{\mathsf{t}}(sk, M, t)$	$\mathsf{Ver}_{\mathsf{t}}(pk, M, \sigma = (\tilde{\sigma}_1, \tilde{\sigma}_2), t)$
$(\mathbb{G}, g, \mathbb{G}_T, p, e) \leftarrow \mathsf{Grp}(1^k)$ $pp := (\mathbb{G}, g, \mathbb{G}_T, p, e)$ $a \leftarrow \mathbb{Z}_p$ $u_0, \ldots, u_m, z, h \leftarrow \mathbb{G}$ $pk := (pp, g^a, u_0, \ldots, u_m, z, h)$ $sk := (pk, a)$ **return** (pk, sk)	$s \leftarrow \mathbb{Z}_p$ $\mathbf{u}_M := u_0 \prod_{i=1}^{m} u_i^{M^i}$ $\tilde{\sigma}_1 := (\mathbf{u}_M)^a (z^t h)^s$ $\tilde{\sigma}_2 := g^s$ **return** $(\tilde{\sigma}_1, \tilde{\sigma}_2)$	**if** $t \notin \{0,1\}^k$ **return** 0 **if** $e(\tilde{\sigma}_1, g) \neq e(\mathbf{u}_M, g^a) e(z^t h, \tilde{\sigma}_2)$ **return** 0 **else** **return** 1

Figure 3.4: The modified Hohenberger-Waters CDH-tag-based signature scheme Σ_t^{CDH} [HW09a].

Verification. $\mathsf{Ver}_{\mathsf{t}}(pk, M, \sigma, t)$, on input pk, message M, purported signature $\sigma = (\tilde{\sigma}_1, \tilde{\sigma}_2)$, and tag t outputs 0 if $t \notin \{0,1\}^k \bigvee e(\tilde{\sigma}_1, g) \neq e(\mathbf{u}_M, g^a) e(z^t h, \tilde{\sigma}_2)$ else 1.

Correctness follows from the bilinearity of the pairing e:

$$\begin{aligned} e(\mathbf{u}_M, g^a) e(z^t h, \tilde{\sigma}_2) &= e(\mathbf{u}_M, g^a) e(z^t h, g^s) = e((\mathbf{u}_M)^a, g) e((z^t h)^s, g) \\ &= e((\mathbf{u}_M)^a (z^t h)^s, g) = e(\tilde{\sigma}_1, g). \end{aligned}$$

Size of Keys and Signature. The public key consists of $(m+4)$ group elements, which is a constant number for a fixed $m \in \mathbb{N}$. The secret key consists of the public key and one element of $\mathbb{Z}_p$. The signature consists of 2 group elements.

Comparison to the Hohenberger and Waters Scheme. The tag-based signature scheme Σ_t^{CDH} described here is derived from the stateful CDH-based scheme of Hohenberger and Waters [HW09a], but with two crucial modifications. In [HW09a] a signature is of the form

$$\begin{aligned} &\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4), \text{ with} \\ &\sigma_1 = (u^M v^r d)^a (w^{\lceil \log(state) \rceil} z^{state} h)^{\tilde{r}}, \sigma_2 = g^{\tilde{r}}, \sigma_3 = r, \sigma_4 = state, \end{aligned}$$

where *state* is a counter, which increments in every single signing process. As mentioned in [HW09a], we can think of $\tilde{r}$ (which we denote as s in our construction) as being the randomness from the Boneh-Boyen selectively-secure Identity-Based-Encryption scheme interpreted as a signature scheme [BB04a]. We will remove all components from this scheme that are not required to prove the scheme EUF-dnaCMA$^*_m$-secure.

First, we substitute the implicit chameleon hash function $u^M v^r$ used in [HW09a] multiplied with a random group element d with a product $\mathbf{u}_M = u_0 \prod_{i=1}^{m} u_i^{M^i}$, which is sufficient for our non-adaptive case and also relevant for EUF-dnaCMA$^*_m$ security since this modification will allow us to simulate up to m signatures per tag t in the proof. You can think of this product as a weak programmable hash function [HJK11a].

Second, we omit the $w^{\lceil \log(state) \rceil}$-factor in the "Boneh-Boyen hash function" which simplifies this part to $(z^t h)^s$, where we use a tag t instead of a state. This factor was relevant if the adversary forges a message with a state it has not queried before, but is not necessary anymore, since the adversary in the EUF-dnaCMA$^*_m$ security experiment has to choose a "recycled" tag $t \in \{t_1, \ldots, t_q\}$ for which he forges a signature.

Here, q is the number of queries the adversary will make and thus is polynomial in k. Let q' be the number of distinct tags queried by F. In the reduction it is possible to guess the forgery tag of the adversary with non-negligible probability $1/q'$.

Theorem 6. If the CDH assumption holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$, then the scheme Σ_t^{CDH} from Figure 3.4 is EUF-dnaCMA$^*_m$-secure. Let F be a PPT adversary with advantage $\varepsilon := \mathsf{Adv}^{\mathsf{euf\text{-}dnacma}^*_{\mathsf{m}}}_{\Sigma_t^{\mathsf{CDH}},F}(k)$ asking for $q := q(k)$ signatures, then it can be used to solve a CDH challenge with probability at least

$$\frac{\varepsilon}{q'},$$

where q' denotes the number of distinct tags queried by F.

Proof. In the following we need to be able to generate valid signatures for the message-tag pairs we receive (non-adaptively at the beginning) from the adversary without knowing the secret key. Therefore we set up the public key in a specific way described below. We obtain q' distinct tags $t_1, \cdots, t_{q'}$. We guess one of them and additionally embed a CDH challenge in the public key which enables to extract a corresponding solution for the problem if the forgery of the adversary is valid and if the forgery tag is equal to our guess. This happens with probability $\frac{\varepsilon}{q'}$.

Public Key Setup. The simulation receives a CDH challenge (g, g^a, g^b), with corresponding group description $(\mathbb{G}, g, \mathbb{G}_T, p, e) \leftarrow \mathsf{Grp}(1^k)$, and pairs $(M_i, t_i)_{i\in[q]}$ for which the adversary F asks for signatures. We first guess an index $i^* \leftarrow [q]$ for which we suppose F will forge a signature on a fresh message $M^* \notin \{M_i\}_{i\in[q]}$ but with forgery tag $t^* = t_{i^*}$.

The adversary F queries $q = \sum_{i=1}^{q'} m_i > 0$ signatures for messages with tags where q' is the number of distinct tags $(t'_i)_{i\in[q']}$ and m_i the number of messages queried for tag t'_i. During the simulation, we denote the messages corresponding to tag t_{i^*} as $M^*_1, \cdots, M^*_{m_{i^*}}$. We construct a polynomial $f \in \mathbb{Z}_p[X]$ such that $f's$ zeros exactly correspond to the messages $M^*_1, \cdots, M^*_{m_{i^*}}$, i.e. $f(M^*_i) = 0$ for $i = 1, \ldots, m_{i^*}$, as follows:

$$f(X) := \prod_{i=1}^{m_{i^*}} (X - M^*_i) = \sum_{i=0}^{m_{i^*}} d_i X^i \in \mathbb{Z}_p[X].$$

The coefficients $d_0, \ldots, d_{m_{i^*}} \in \mathbb{Z}_p$ are efficiently computable. In particular, for $m_{i^*} = 0$ we have $\prod_{i=1}^{0}(X - M^*_i) = 1$.

Using the above coefficients, we set up the public key for F by first choosing random $R_0, \ldots, R_m, x_z, x_h \in \mathbb{Z}_p$ and then set

$$\begin{aligned} u_i &:= \begin{cases} (g^b)^{d_i} g^{R_i}, & i = 0, \ldots, m_{i^*} \\ g^{R_i} & i = m_{i^*}+1, \cdots, m \end{cases} \\ z &:= g^b g^{x_z}, \\ h &:= g^{-bt_{i^*}} g^{x_h} \end{aligned}$$

embedding g^b from its CDH challenge in each group element and the specific tag t_{i^*} in the group element h.

The simulation sets

$$pp := (\mathbb{G}, g, \mathbb{G}_T, p, e)$$

and sends to the adversary F

$$pk := (pp, g^a, u_0, \ldots, u_m, z, h)$$

and implicitly sets the secret key as

$$sk := a.$$

By defining

$$R(X) := \sum_{i=0}^{m} R_i X^i$$

we can write

$$\begin{aligned}\mathbf{u}_M &= u_0 \prod_{i=1}^{m} u_i^{M^i} \\ &= g^{b\cdot d_0 + R_0 + \sum_{i=1}^{m_{i^*}} M^i \cdot b \cdot d_i + \sum_{i=1}^{m} M^i \cdot R_i} \\ &= g^{b\cdot(\sum_{i=0}^{m_{i^*}} d_i \cdot M^i) + \sum_{i=0}^{m} R_i \cdot M^i} \\ &= g^{bf(M)+R(M)}\end{aligned}$$

Signing. There are two cases we have to consider when F non-adaptively asks for a signature for (M_i, t_i).
If $t_i = t_{i^*}$ and thus $M_i = M_j^*$ for some $j = 1, \ldots, m_{i^*}$, we are able to compute a valid signature as follows: We choose a random $s_i \leftarrow \mathbb{Z}_p$ and set $\sigma_i := (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i})$, where

$$\begin{aligned}\tilde{\sigma}_{1,i} &= (g^a)^{R(M_j^*)} \cdot (z^{t_{i^*}} h)^{s_i} \ , \\ \tilde{\sigma}_{2,i} &= g^{s_i}.\end{aligned}$$

This is justified by the fact that $f(M_j^*) = 0$ and thus $g^{bf(M_j^*)} = 1$. Hence, we have

$$\begin{aligned}\tilde{\sigma}_{1,i} &= (g^a)^{R(M_j^*)} \cdot (z^{t_{i^*}} h)^{s_i} \\ &= (g^{bf(M_j^*)} g^{R(M_j^*)})^a \cdot (z^{t_{i^*}} h)^{s_i} \\ &= (\mathbf{u}_{M_j^*})^a \cdot (z^{t_{i^*}} h)^{s_i}.\end{aligned}$$

Therefore, we have

$$\begin{aligned}e(\tilde{\sigma}_{1,i}, g) &= e((\mathbf{u}_{M_j^*})^a \cdot (z^{t_{i^*}} h)^{s_i}, g) \\ &= e(\mathbf{u}_{M_j^*}, g^a) e(z^{t_{i^*}} h, g^{s_i}) \\ &= e(\mathbf{u}_{M_j^*}, g^a) e(z^{t_{i^*}} h, \tilde{\sigma}_{2,i})\end{aligned}$$

resulting in $\sigma_i = (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i})$ being a valid signature for (M_i, t_i) (Figure 3.4).

If $t_i \neq t_{i^*}$ then, following the original Boneh-Boyen simulation, we choose a random $s_i' \leftarrow \mathbb{Z}_p$ and set

$$\begin{aligned}S_i :&= g^{s_i'} / (g^a)^{f(M_i)(t_i - t_{i^*})^{-1}} \\ &= g^{s_i' - a f(M_i)(t_i - t_{i^*})^{-1}}.\end{aligned}$$

We compute the corresponding signature $\sigma_i := (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i})$ as follows:

$$\begin{aligned}\tilde{\sigma}_{1,i} &= (g^a)^{R(M_i)} \cdot S_i^{x_z t_i + x_h} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \ , \\ \tilde{\sigma}_{2,i} &= S_i.\end{aligned}$$

Thus, implicitly, we set the randomness $s_i = s_i' - f(M_i)(t_i - t_{i^*})^{-1} \bmod p$ and obtain $S_i = g^{s_i}$.

Due to the following identity $\sigma_i = (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i})$ is valid

$$\begin{aligned}
\tilde{\sigma}_{1,i} &= (g^a)^{R(M_i)} \cdot S_i^{x_z t_i + x_h} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \\
&= (g^{R(M_i)})^a \cdot (g^{x_z t_i} g^{x_h})^{s_i} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \\
&= (g^{ab})^{f(M_i)} \cdot (g^{ab})^{-f(M_i)} \cdot (g^{R(M_i)})^a \cdot (g^{x_z t_i} g^{x_h})^{s_i} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \\
&= (g^{ab})^{f(M_i)} \cdot (g^{R(M_i)})^a \cdot (g^{x_z t_i} g^{x_h})^{s_i} \cdot (g^b)^{s_i'(t_i - t_{i^*})} \cdot (g^{-ab})^{f(M_i)} \\
&= (g^{ab})^{f(M_i)} \cdot (g^{R(M_i)})^a \cdot (g^{x_z t_i} g^{x_h})^{s_i} \cdot (g^{b(t_i - t_{i^*})})^{\overbrace{s_i' - af(M_i)(t_i - t_{i^*})^{-1}}^{s_i}} \\
&= ((g^{bf(M_i) + R(M_i)})^a \cdot (g^{x_z t_i} g^{x_h})^{s_i} \cdot (g^{b(t_i - t_{i^*})})^{s_i} \\
&= (\mathbf{u}_{M_i})^a \cdot (g^{x_z t_i} \cdot g^{bt_i} \cdot g^{x_h} \cdot g^{-bt_{i^*}})^{s_i} \\
&= (\mathbf{u}_{M_i})^a \cdot ((g^{b+x_z})^{t_i} \cdot (g^{-bt_{i^*} + x_h}))^{s_i} \\
&= (\mathbf{u}_{M_i})^a \cdot (z^{t_i} h)^{s_i}.
\end{aligned}$$

Extract from Forgery. After receiving the public key pk and the signatures $(\sigma_i)_{i \in [q]}$, the adversary F responds with (M^*, σ^*, t^*) for some tag $t^* \in \{t_1, \ldots, t_{q'}\}$ and $\sigma^* = (\tilde{\sigma}_1^*, \tilde{\sigma}_2^*)$. We abort if σ^* is not a valid forgery or any other winning condition is violated. Otherwise, since the verification equation holds, we have

$$\begin{aligned}
\tilde{\sigma}_1^* &= (\mathbf{u}_{M^*})^a (z^{t^*} h)^{s^*} , \\
\tilde{\sigma}_2^* &= g^{s^*}.
\end{aligned}$$

for some suitable s^*.

If $t_{i^*} \neq t^*$, we abort. Otherwise, our guess was correct ($t_{i^*} = t^*$) and it holds that

$$\begin{aligned}
\tilde{\sigma}_1^* &= (\mathbf{u}_{M^*})^a (z^{t^*} h)^{s^*} \\
&= ((g^b)^{f(M^*)} (g^{R(M^*)}))^a ((g^{b+x_z})^{t^*} (g^{-bt_{i^*} + x_h}))^{s^*} \\
&\overset{(*)}{=} g^{abf(M^*)} g^{aR(M^*)} (g^{x_z t^*} g^{x_h})^{s^*} \\
&= g^{abf(M^*)} g^{aR(M^*)} g^{s^*(x_z t^* + x_h)},
\end{aligned}$$

where $(*)$ follows because $t_{i^*} = t^*$. Since σ^* is a valid forgery, we have $M^* \notin \{M_1^*, \ldots, M_{m_{i^*}}^*\}$. Thus we have $f(M^*) \neq 0$ and the simulator is able to compute

$$R(M^*), \text{ and thus } (g^a)^{R(M^*)}$$

$$x_z t^* + x_h, \text{ and thus } \tilde{\sigma}_2^{*(x_z t^* + x_h)}$$

$$f(M^*)^{-1} \text{ in } \mathbb{Z}_p.$$

Hence, the following computation yields g^{ab} :

$$\begin{aligned}
(\tilde{\sigma}_1^* / (g^{aR(M^*)} \tilde{\sigma}_2^{*(x_z t^* + x_h)}))^{f(M^*)^{-1}} &= (g^{abf(M^*) + aR(M^*) + s^*(x_z t^* + x_h)} \cdot g^{-aR(M^*) - s^*(x_z t^* + x_h)})^{f(M^*)^{-1}} \\
&= (g^{abf(M^*)})^{f(M^*)^{-1}} \\
&= g^{ab}.
\end{aligned}$$

Analysis. We show that the adversary F cannot distinguish between the experiment and the simulation. By ε we denote the advantage of the adversary F in the experiment and by success the event, that the simulation outputs a solution g^{ab}. The simulator does not pick $(u_i)_{i=0}^m$, z, and h at random, but sets them as described above. Since the R_i, x_z and x_h are chosen randomly and independently, this yields the correct distribution, so the view of the adversary is

still identical to his view in the experiment and the signatures are correctly distributed. Hence, the simulator is successful if it does not abort and if F is successful. Further, if F is successful, an abort only occurs if t^* is not guessed correctly (out of q' possibilities). Thus, we have

$$\Pr[\mathsf{success}] = \Pr[F \text{ successful}] \cdot \Pr[t_{i^*} = t^*] = \frac{\varepsilon}{q'}.$$

□

3.4.2 Full EUF-CMA-Security and Optimizations

Now, we can use the generic transformation from Theorem 2 and our result from Theorem 6 to construct a stateless EUF-dnaCMA secure signature scheme. To achieve a fully EUF-CMA-secure signature scheme, we add a chameleon hash function (see Lemma 5). Additionally, we use aggregation to optimize the length of the signatures. We first give a description of the optimized CDH-based scheme $\Sigma^{\mathsf{CDH}}_{\mathsf{opt}}$ and explain these steps in more detail afterwards.

We construct our optimized CDH-based signature scheme $\Sigma^{\mathsf{CDH}}_{\mathsf{opt}} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ with message space $\mathcal{M} = \mathbb{Z}_p$, a pseudorandom function $\mathsf{PRF}^{\mathcal{T}_i}_{\kappa}$ with range $\mathcal{T}_k = \{0,1\}^k$ for generating internal tags t_i from corresponding tag space $\mathcal{T}_i = \{0,1\}^{\lceil c^i \rceil}$ and $l := \lfloor \log_c(k) \rfloor$ instances as described in section 3.3 as follows (see Figure 3.5):

Key Generation. $\mathsf{Gen}(1^k)$, on input 1^k, runs $(\mathbb{G}, g, \mathbb{G}_T, p, e) \leftarrow \mathsf{Grp}(1^k)$, samples a random exponent $a \leftarrow \mathbb{Z}_p$, random group elements $w, u_0, \ldots, u_m, z_1, \ldots, z_l, h \leftarrow \mathbb{G}$, a random $\kappa \in \{0,1\}^k$, and outputs $(pk, sk) := ((\mathbb{G}, g, \mathbb{G}_T, p, e, w, \kappa, g^a, u_0, \ldots, u_m, z, h), (pk, a))$.

Sign. $\mathsf{Sig}(sk, M)$, on input sk and message $M \in \mathbb{Z}_p$, samples $s, r \leftarrow \mathbb{Z}_p$, computes $x := g^M w^r = \mathsf{CH}_{(g,w)}(M, r)$, $\mathbf{u}_x := u_0 \prod_{i=1}^{m} u_i^{x^i}$, $t_i := \mathsf{PRF}^{\mathcal{T}_i}_{\kappa}(x)$ for $i = 1, \ldots, l$, and $\mathbf{z}_t := \prod_{i=1}^{l} z_i^{t_i}$ and outputs $(\tilde{\sigma}_1, \tilde{\sigma}_2, r) := ((\mathbf{u}_x)^a (\mathbf{z}_t \cdot h)^s, g^s, r)$.

Verification. $\mathsf{Ver}(pk, M, \sigma)$, on input pk, message M, purported signature $\sigma = (\tilde{\sigma}_1, \tilde{\sigma}_2, r)$, computes $x := \mathsf{CH}_{(g,w)}(M, r)$, and $t_i := \mathsf{PRF}^{\mathcal{T}_i}_{\kappa}(x)$, for $i = 1, \ldots, l$, and outputs 1 if $e(\tilde{\sigma}_1, g) = e(\mathbf{u}_x, g^a) e(h \prod_{i=1}^{l} z_i^{t_i}, \tilde{\sigma}_2)$, else 0.

Correctness follows again from the bilinearity of the pairing e:

$$\begin{aligned} e(\mathbf{u}_x, g^a) e(h \prod_{i=1}^{l} z_i^{t_i}, \tilde{\sigma}_2) &= e(\mathbf{u}_x, g^a) e(\mathbf{z}_t \cdot h, g^s) = e((\mathbf{u}_x)^a, g) e((\mathbf{z}_t \cdot h)^s, g) \\ &= e((\mathbf{u}_x)^a (\mathbf{z}_t \cdot h)^s, g) = e(\tilde{\sigma}_1, g). \end{aligned}$$

Additional Group Elements in the Public Key. The public key pk of our optimized CDH-based scheme $\Sigma^{\mathsf{CDH}}_{\mathsf{opt}}$ consists of additional group elements $w, z_1, \ldots, z_l$ compared to our CDH-based scheme Σ^{CDH}. The group element w is necessary to implement the chameleon hash function $\mathsf{CH}_{(g,w)}(M, r) := g^M w^r$ from section 3.2, where the trapdoor τ is the exponent $\log_g(w)$. We recall, that given a valid signature $\sigma = (\tilde{\sigma_1}, \tilde{\sigma_2}, r)$ for a message M with randomness r, it is possible to use the trapdoor of the chameleon hash function to compute a randomness $r' = \mathsf{CHTrapColl}(\tau, M, r, M')$ for a message M' such that $\sigma' = (\tilde{\sigma_1}, \tilde{\sigma_2}, r')$ is a valid signature for M'. Note that this new signature is computable without knowing the secret key $sk = a$ but only by knowing the trapdoor of the chameleon hash function and a valid signature for M. This is a useful tool to convert EUF-dnaCMA-secure signature schemes to EUF-CMA-secure signature schemes (see also Lemma 5).

Furthermore, the group elements $z_1, \ldots, z_l$ are necessary for aggregation: In the generic transformation we used $\lfloor \log_c(k) \rfloor$ instances of an EUF-dnaCMA$^*_m$-secure tag-based scheme to achieve

$\mathsf{Gen}(1^k)$	$\mathsf{Sig}(sk, M)$	$\mathsf{Ver}(pk, M, \sigma = (\tilde{\sigma}_1, \tilde{\sigma}_2, r))$

$(\mathbb{G}, g, \mathbb{G}_T, p, e) \leftarrow \mathsf{Grp}(1^k)$
$a \leftarrow \mathbb{Z}_p$
$w, u_0, \ldots, u_m,$
$z_1, \ldots, z_l, h \leftarrow \mathbb{G}$
$\kappa \leftarrow \{0,1\}^k$
$pp := (\mathbb{G}, g, \mathbb{G}_T, p, e, w, \kappa)$
$pk := (pp, g^a, u_0, \ldots, u_m,$
$z_1, \ldots, z_l, h)$
$sk := (pk, a)$
return (pk, sk)
	$s, r \leftarrow \mathbb{Z}_p$
	$g^M w^r = \mathsf{CH}_{(g,w)}(M, r)$
	$x := g^M w^r$
	$\mathbf{u}_x := u_0 \prod_{i=1}^m u_i^{x^i}$
	for $i := 1$ **to** l **do**
	$t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x)$
	$\mathbf{z}_t := \prod_{i=1}^l z_i^{t_i}$
	$\tilde{\sigma}_1 := (\mathbf{u}_x)^a (\mathbf{z}_t \cdot h)^s$
	$\tilde{\sigma}_2 := g^s$
	return $(\tilde{\sigma}_1, \tilde{\sigma}_2, r)$
		$x := \mathsf{CH}_{(g,w)}(M, r)$
		for $i := 1$ **to** l **do**
		$t_i := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x)$
		if $e(\mathbf{u}_x, g^a) e(h \prod_{i=1}^l z_i^{t_i}, \tilde{\sigma}_2)$
		$= e(\tilde{\sigma}_1, g)$
		return 1
		else
		return 0

Figure 3.5: The optimized CDH-based signature scheme $\Sigma_{\mathsf{opt}}^{\mathsf{CDH}}$.

an EUF-dnaCMA-secure scheme. Adding a chameleon hash function for each instance already achieves EUF-CMA security. However, this would result in signatures comprising $\mathbf{O}(\log k)$ group elements of the form $\sigma = (\sigma_1, \ldots, \sigma_{\log k})$, where $\sigma_i = (\tilde{\sigma}_{1,i}, \tilde{\sigma}_{2,i}, r_i)$. To improve this and achieve constant size signatures, we use aggregation, i.e. we essentially multiply the signatures of each instance similar to [LOSSW06]. In particular, an aggregated signature in our optimized scheme is of the form $\sigma = (\tilde{\sigma}_1, \tilde{\sigma}_2, r) = ((\mathbf{u}_x)^a (\mathbf{z}_t \cdot h)^s, g^s, r)$, where $x = g^M w^r$ is the chameleon hash value, i.e we replace the term z^t with a product $\mathbf{z}_t := \prod_{i=1}^l z_i^{t_i}$. Despite the use of aggregation, we can re-use $u_0, \ldots, u_m, h$, one $sk := a$ and one randomness s for all instances i. This is due to the fact that during the reduction the simulation is able to set up these elements as before and it is only necessary to use distinct z_i, since the simulation embeds the CDH challenge only in one specific z_{i^*} such that it is still possible to generate valid signatures but also to extract a solution from the forgery signature.

Unfortunately, to account for constant-size signatures, our public key consists now of $\mathbf{O}(\log k)$ group elements. In this sense our optimization is rather a tradeoff: We prefer constant-size signatures with public keys of logarithmic-size over logarithmic-size signatures with constant-size public keys. Public keys are transmitted only once, whereas signatures are transmitted more often.

Theorem 7. If the CDH assumption holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$, then the optimized CDH-based signature scheme $\Sigma_{\mathsf{opt}}^{\mathsf{CDH}}$ in Figure 3.5 is an EUF-CMA-secure signature scheme. Let F be a PPT adversary with advantage $\varepsilon := \mathsf{Adv}_{\Sigma_{\mathsf{opt}}^{\mathsf{CDH}}, F}^{\mathsf{euf\text{-}cma}}(k)$ asking for $q := q(k)$ signatures, then it can be used to solve a CDH challenge with probability at least

$$\frac{\varepsilon^{c/m+1} - 2\varepsilon^{c/m}(\varepsilon_{\mathsf{PRF}} + \varepsilon_{\mathsf{CH}})}{2^{2+c/m} \cdot q^{c+c/m}},$$

where $\varepsilon_{\mathsf{PRF}}$ and $\varepsilon_{\mathsf{CH}}$ correspond to the advantages for breaking PRF and CH, respectively.

Proof. We only sketch the proof here, because it is essentially a combination of Theorem 2, Lemma 4, Lemma 5 and the proof from Theorem 6. We emphasize the differences and important parts. In particular, we have to deal with $l = \lfloor \log_c(k) \rfloor$ instances and obtain our signature by generic aggregation.

Public Key Setup. First, we select an index i^*, as in Lemma 4, and guess a tag t_{i^*} from the corresponding set $\mathcal{T}_{i^*}$. Next, we pick a random $\kappa \leftarrow \{0,1\}^k$, a random $\tau \leftarrow \mathbb{Z}_p$ and set $w := g^\tau$.

Thus, τ is a trapdoor for the chameleon hash function $\mathsf{CH}_{(g,w)}$. Then, we sample random distinct $M'_1, \ldots, M'_q, r'_1, \ldots, r'_q \leftarrow \mathbb{Z}_p$ and compute the chameleon hash values

$$x_j = \mathsf{CH}_{(g,w)}(M'_j, r'_j),$$

for message M'_j and corresponding randomness r'_j, $j \in [q]$.[1] Due to the collision resistance of the chameleon hash function and the fact that $|\mathbb{Z}_p| >> q$, we can assume that all x_j are distinct, otherwise we abort. We derive tags

$$t_i^{(j)} := \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x_j)$$

for each instance $i \in [l]$ and each query $j \in [q]$. We recall, for message M'_j, we have corresponding tags $t_1^{(j)}, \ldots, t_l^{(j)}$ for each instance.

Then, we consider the set $J := \{\, j \in [q] \mid t_{i^*}^{(j)} = t_{i^*} \}$. Hence, $|J|$ is the number of all messages M'_j, which corresponding tag $t_{i^*}^{(j)}$ used for instance i^* is the same as our guessed tag t_{i^*}. If $|J| > m$, we abort, otherwise $|J| =: m_{i^*} \leq m$. Similar to the proof of Theorem 6, we define a polynomial $f \in \mathbb{Z}_p[X]$ s.t. $f(x_j) = 0$ iff $j \in J$. Hence, f is of the form

$$f(X) := \prod_{j \in J} (X - x_j) = \sum_{i=0}^{m_{i^*}} d_i X^i,$$

for appropriate coefficients $d_0, \ldots, d_{m_{i^*}} \in \mathbb{Z}_p$.

We sample $u_0, \ldots, u_m, h$ as in the proof of Theorem 6 to embed our CDH challenge. Further, we choose random group elements z_i for $i \in [l]$ by choosing random exponents $x_{z_i} \in \mathbb{Z}_p$ for them and additionally embed the CDH challenge in z_{i^*}, i.e.:

$$u_i := \begin{cases} (g^b)^{d_i} g^{R_i}, & i = 0, \ldots, m_{i^*} \\ g^{R_i} & i = m_{i^*} + 1, \cdots, m \end{cases}$$

$$z_i := \begin{cases} g^{x_{z_i}}, & i \neq i^* \\ g^b g^{x_{z_{i^*}}}, & i = i^* \end{cases}$$

$$h := g^{-b t_{i^*}} g^{x_h}.$$

The simulation sets

$$pp := (\mathbb{G}, g, \mathbb{G}_T, p, e, w, \kappa),$$

and sends to the adversary F

$$pk := (pp, g^a, u_0, \ldots, u_m, z_1, \ldots, z_l, h)$$

and, hence, implicitly sets the secret key as

$$sk := a.$$

By defining $R(X) := \sum_{i=0}^{m} R_i X^i$, we can write

$$\mathbf{u}_x = g^{bf(x)+R(x)}.$$

Signing. After receiving the public key pk, the adversary adaptively queries messages $M_1, \ldots, M_q$. The simulation has to be able to produce correctly distributed valid signatures $\sigma^{(j)} = (\tilde{\sigma}_{1,j}, \tilde{\sigma}_{2,j}, r_j)$

[1] Here, we assume to know the number of signatures $q \geq 0$ the adversary F is going to query.

for each of them. To do so, we first compute r_j s.t. $x_j = \mathsf{CH}_{(g,w)}(M_j, r_j)$, by using the trapdoor τ. Thus, the tags $t_i^{(j)} = \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x_j)$ belong to M_j, for $i = 1, \ldots, l$.

Now, we consider the instance $i = i^*$ for each query: Again, we have two cases, either $t_{i^*}^{(j)} = t_{i^*}$ or $t_{i^*}^{(j)} \neq t_{i^*}$. In both cases we can apply the same techniques as in the proof of Theorem 6 to obtain an appropriate $(\mathbf{u}_{x_j})^a (z_{i^*}^{t_{i^*}^{(j)}} h)^{s_j}$.

For the other instances $i \neq i^*$, we compute $(z_i^{t_i^{(j)}})^{s_j} = (g^{x_{z_i} t_i^{(j)}})^{s_j} = (g^{s_j})^{x_{z_i} t_i^{(j)}}$ due to the fact that we know the exponents x_{z_i}.

Then we can aggregate all z_i, $i = 1, \ldots, l$, by muliplying them together, to obtain a valid signature for each query $M_j, j \in [q]$, as follows:

$$\begin{aligned}
\tilde{\sigma}_{1,j} &= (\mathbf{u}_{x_j})^a (h \prod_{i=1}^{l} z_i^{t_i^{(j)}})^{s_j}, \\
\tilde{\sigma}_{2,j} &= g^{s_j}, \\
\sigma^{(j)} &:= ((\mathbf{u}_{x_j})^a (h \prod_{i=1}^{l} z_i^{t_i^{(j)}})^{s_j}, g^{s_j}, r_j) = (\tilde{\sigma}_{1,j}, \tilde{\sigma}_{2,j}, r_j).
\end{aligned}$$

Extract from Forgery. Finally, the adversary F responds with (M^*, σ^*) where $\sigma^* = (\tilde{\sigma}_1^*, \tilde{\sigma}_2^*, r^*)$. We abort if σ^* is not valid or M^* is not fresh, i.e. $M^* = M_j$, for some $j \in [q]$. We can assume that F has not produced a collision $x^* = \mathsf{CH}_{(g,w)}(M^*, r^*) = x_j$ for some $j \in [q]$. The probability of that is negligible due to the collision resistance of the chameleon hash function. Thus, we have $f(x^*) \neq 0$. We compute $t_i^* = \mathsf{PRF}_\kappa^{\mathcal{T}_i}(x^*)$ for each instance $i \in [l]$. If $t_{i^*}^* \neq t_{i^*}$, we guessed t_{i^*} incorrectly and abort, otherwise since the verification equation holds we have

$$\begin{aligned}
\tilde{\sigma}_1^* &= (\mathbf{u}_{x^*})^a (h \prod_{i=1}^{l} z_i^{t_i^*})^{s^*}, \\
\tilde{\sigma}_2^* &= g^{s^*}
\end{aligned}$$

for some suitable s^*.

We can compute $(g^{s^*})^{x_{z_i} t_i^*} = (z_i^{t_i^*})^{s^*}$ for $i \neq i^*$ and obtain

$$\sigma_1^* / \prod_{i=1, i \neq i^*}^{l} (z_i^{t_i^*})^{s^*} = (\mathbf{u}_{x^*})^a (z_{i^*}^{t_{i^*}^*} h)^{s^*}.$$

Since $f(x^*) \neq 0$ and $t_{i^*}^* = t_{i^*}$ we are able to extract a solution g^{ab} to the CDH challenge as follows:

$$\begin{aligned}
(\mathbf{u}_{x^*})^a (z_{i^*}^{t_{i^*}^*} h)^{s^*} &= ((g^b)^{f(x^*)} g^{R(x^*)})^a ((g^{b + x_{z_{i^*}}})^{t_{i^*}^*} (g^{-b t_{i^*} + x_h}))^{s^*} \\
&\overset{(*)}{=} ((g^b)^{f(x^*)} g^{R(x^*)})^a (g^{b t_{i^*}^* + x_{z_{i^*}} t_{i^*}^* - b t_{i^*}^* + x_h})^{s^*} \\
&= g^{abf(x^*)} g^{aR(x^*)} (g^{x_{z_{i^*}} t_{i^*}^*} g^{x_h})^{s^*} \\
&= g^{abf(x^*)} g^{aR(x^*)} g^{s^*(x_{z_{i^*}} t_{i^*}^* + x_h)}.
\end{aligned}$$

where $(*)$ follows because of $t_{i^*}^* = t_{i^*}$. Thus

$$(g^{abf(x^*)} g^{aR(x^*)} g^{s^*(x_{z_{i^*}} t_{i^*}^* + x_h)} / (g^{aR(x^*)} \tilde{\sigma}_2^{*(x_{z_{i^*}} t_{i^*}^* + x_h)}))^{f(x^*)^{-1}} = g^{ab}.$$

Analysis. The analysis is similar to Theorem 2, Lemma 4, Lemma 5 and the proof of Theorem 6. We denote by ε the advantage of the adversary F in the experiment and by $\mathsf{success}$ the event

that the simulation outputs a solution g^{ab}. Hence,

$$\Pr[\mathsf{success}] \geq \Pr[t_{i^*} = t^*]\left(\frac{\varepsilon}{2} - (\varepsilon_{\mathsf{PRF}} + \varepsilon_{\mathsf{CH}})\right) = \frac{1}{|\mathcal{T}_{i^*}|}\left(\frac{\varepsilon}{2} - (\varepsilon_{\mathsf{PRF}} + \varepsilon_{\mathsf{CH}})\right)$$
$$\overset{(*)}{\geq} \frac{\varepsilon^{1+c/m} - 2\varepsilon^{c/m}(\varepsilon_{\mathsf{PRF}} + \varepsilon_{\mathsf{CH}})}{2^{2+c/m} \cdot q^{c+c/m}},$$

where $(*)$ holds by Lemma 4, since we have $|\mathcal{T}_{i^*}| \leq 2 \cdot \left(\frac{2q^{m+1}}{\varepsilon(k)}\right)^{c/m}$. Here, $\varepsilon_{\mathsf{PRF}}$ is the advantage for a suitable adversary on PRF and $\varepsilon_{\mathsf{CH}}$ is the advantage to produce a collision for CH, both being negligible in the security parameter. □

Remark 1. We want to clarify once again, why we have to use $\mathbf{O}(\log(k))$ group elements in the public key. In the security proof we have to set up the public key elements in a way that enables to generate valid signatures and to extract a solution g^{ab} of the CDH challenge from the forgery of the adversary. If we would re-use the group element z for aggregation instead of different z_i's our product would then be of the form

$$\mathbf{z}_t = \prod_{i=1}^{l} z^{t_i} = z^{\sum_{i=1}^{l} t_i}.$$

In the security proof we embed the CDH challenge in one specific $z_{i^*} = g^b g^{x_{z_{i^*}}}$ tailored to t_{i^*} and set $z_i = g^{x_{z_i}}$, for $i \neq i^*$. Now, we would have to embed the challenge in $z = g^b g^{x_z}$, which results in

$$\mathbf{z}_t = z^{\sum_{i=1}^{l} t_i} = g^{(b+x_z)\sum_{i=1}^{l} t_i}$$

After receiving a valid forgery $\sigma^* = (\tilde{\sigma}_1^*, g^{s^*}, r^*)$ from the adversary, we are not able to compute a solution g^{ab}, anymore, since we need to compute

$$(z^{\sum_{i \neq i^*} t_i})^{s^*} = (g^{(b+x_z)\sum_{i \neq i^*} t_i})^{s^*}$$

for that. Thus, we have to use either b or s^* to compute

$$(g^b)^{s^*} = (g^{s^*})^b$$

which we both do not know.

3.5 State-of-the-Art

In [BHJKS13; BHJKSS13] further instantiations based on the RSA and Short Integer Solutions (SIS) assumption, respectively, are given. These are not part of this thesis, but will be briefly mentioned for completeness.

3.5.1 RSA- and SIS-Based Instantiations

With the generic transformation from section 3.3 it is not only possible to construct EUF-CMA-secure digital signatures under the CDH assumption (see section 3.4) but also under the RSA and SIS assumption. For RSA, similar to CDH, further optimizations on the system's parameters (e.g. aggregation of signatures) can be done (see [BHJKS13; BHJKSS13]). Both public key and signatures consist of $\mathbf{O}(1)$ elements. This yields to the most efficient (also in terms of computation time) fully secure RSA-based scheme known at that time.

In the SIS-based scheme, signatures consist of $\mathbf{O}(\log(k) \cdot m)$ and verification keys of $\mathbf{O}(n \cdot m)$ group elements, where n, m denote the usual SIS matrix dimensions. Compared to SIS-based schemes at that time, this gives very small public keys, at the price of slightly larger signatures. This is due to the fact that aggregation techniques for lattice-based signatures rarely existed at that time.

3.5.2 Recent Improvements

In [XQZL14] based on the confined guessing technique, especially for the lattice-based signature scheme, they construct a lattice-based tag-based signature scheme with shorter signature length proven strongly secure in the standard model. They reduce the size of the signature by adopting the technique of lattice basis delegation with fixed dimension. Shortly after that [DM14] combines the confined guessing technique with the "vanishing trapdoors" technique of Boyen [Boy10] and achieves short signatures consisting only of a single lattice vector but with larger public keys. This has been improved to short public keys and short signatures in [AS15]. But all of these works, including ours, have a rather large security loss in the reduction. The latest work from Boyen and Li [BL16] and Alperin-Sheriff and Apon [ASA17] for the first time constructed lattice-based signature schemes with short signatures and rather efficient security reductions. In [BL16] they have an almost tight security reduction which was improved in [ASA17] to a tight security reduction. However, both still require large public keys.

Concerning tight security reductions Kajita et al. [KOF17] present a signature scheme with the tightest security reduction among known constant-size signature schemes secure under the CDH assumption. They first construct a signature scheme, satisfying a new security notion, denoted as existentially unforgeable against extended random-message attacks (EUF-XRMA), based on our construction but with a tighter security reduction. They transform it to an EUF-CMA-secure scheme without loosing the tightness.

4 A Fault-Tolerant Aggregate Signature Scheme

An aggregate signature scheme is a variant of a digital signature scheme which allows to compress a big amount of individual signatures from different signers into one short signature, referred to as the *aggregate signature* [BGLS03]. This is very useful for saving verification computations, bandwidth and storage space. Therefore, aggregate signatures have plenty of applications [AGH10].

For example, a well-known field of application are *sensor networks* [CCMT09], which consist of several small sensors that measure an aspect of their physical environment and send their findings to a central base station. Digital signatures ensure the integrity and authenticity of the measurements during transfer from the sensors to the base station. Using a conventional digital signature scheme, the verifying base station would need to receive each signature separately, which is bandwidth-intensive. However, if the signatures were aggregated beforehand using an aggregate signature scheme, the bandwidth consumption on the side of the base station is reduced drastically. Also, verifying an aggregate signature is typically considerably faster than verifying all individual signatures.

However, if the aggregate signature contains only one invalid individual signature, the verification algorithm outputs *invalid*. Moreover, there is no way to identify which individual signatures are invalid or which individual signatures are still valid.

To overcome this problem, we construct the first fault-tolerant aggregate signature scheme, published in [HKKKR16]. A fault-tolerant aggregate signature scheme extends an aggregate signature scheme, such that the verification algorithm does not only output *valid* or *invalid* but instead the list of correctly signed messages belonging to an aggregate signature. That is desirable for many applications, since an invalid aggregate signature results in re-computing and re-sending all data.

We emphasize here, that we almost entirely taken the important parts of these chapter from [HKKKR16], partly verbatim, with some useful additions, descriptions and further explanations.

Our Contribution

We construct the first d-fault-tolerant aggregate signature scheme based on d-cover-free families, which are a combinatorial structure related to error-correcting codes and are able to "handle" up to d errors. In a d-fault-tolerant aggregate signature scheme the verification algorithm outputs a list of correctly signed messages instead of only *valid* or *invalid*. This prevents re-computations and re-sending of large amounts of data. Thus, we define a new cryptographic building block and show an instantiation with respect to a concrete d-cover-free family based on polynomials over a finite field.

4.1 Organization

In section 4.2 we give some necessary definitions and notations and repeat the notion of aggregate signatures. We discuss the notion of fault-tolerance in section 4.3. The verification algorithm of our fault-tolerant aggregate signature scheme is able to identify the subset of all messages belonging to an aggregate that were signed correctly, provided that the number of aggregated faulty signatures does not exceed a certain bound.

In section 4.4 we give a construction of a d-fault-tolerant aggregate signature scheme from an ordinary aggregate signature scheme based on a d-cover-free family [Für96], where d denotes the number of faulty individual signatures, the scheme can cope with. An aggregate signature of our scheme is a vector of aggregated signatures of the underlying scheme. The maximum number n of signatures that can be aggregated in our scheme must be fixed in advance, and thus, is bounded. The size of the vector, i.e. of our aggregate signature, is logarithmic in n.

We also show an unbounded construction in section 4.4.1, where the length of the aggregate signature grows linearly in the number of aggregated signatures, but the factor in this linear function can be made arbitrarily small.

In section 4.4.2 we present an additional feature of our scheme, denoted as selective verification. Selective verification enables to verify some individual message-signature pair in an aggregate signature without verifying the whole aggregate, which speeds up verification.

In section 4.5 we give a concrete instantiation of a d-cover-free family which is based on polynomials over a finite field [KRS99]. We generalize this to multivariate polynomials in section 4.5.3.

In section 4.6.1 we mention a useful application of our d-fault-tolerant aggregate signature scheme in the secure logging scenario, published in [HKKKH17], which is not part of this thesis.

4.2 Preliminaries

In this section we will give some necessary existing definitions and notations.

Definition 16 (Multiset, Multiplicity)**.** A multiset is a pair (T, μ), where T is a set and $\mu : T \to \mathbb{N}_{>0}$ is a mapping to the strictly positive natural numbers. For $t \in T$ the natural number $\mu(t)$ is the *multiplicity* (number of occurrence) of t in T. In the following we will omit the mapping μ, when using multisets.

For two multisets T_1, T_2, the union $T_1 \cup T_2$ is defined as the multiset where the multiplicity of each element in $T_1 \cup T_2$ is the sum of the multiplicities in T_1, T_2.

Remark 2. We consider multisets which consists of public key and message pairs, e.g. $T = ((pk_1, M_1), \ldots, (pk_n, M_n))$. Note, that the public keys and messages are not necessarily distinct.

Aggregate Signatures. We quickly review the definition of aggregate signature schemes and the associated security notion, defined in [BGLS03].

Definition 17. An *aggregate signature scheme* $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Agg}, \mathsf{Verify})$ with message space $\mathcal{M}_k$ consists of four PPT algorithms:

Key Generation. $\mathsf{Gen}(1^k)$, on input 1^k, outputs a key pair (pk, sk).

Sign. $\mathsf{Sig}(sk, M)$, on input a secret key sk and a message M, outputs a signature σ.

Aggregation $\mathsf{Agg}(C_1, C_2, \tau_1, \tau_2)$, on input two multisets of public-key and message pairs C_1 and C_2 and corresponding (aggregate) signatures τ_1 and τ_2, outputs (C, τ), where τ is an aggregate signature, certifying the validity of the messages in $C := C_1 \cup C_2$ under the corresponding public keys.

(Aggregate) Verification. $\mathsf{Verify}(C, \tau)$, on input a multiset of public key and message pairs C and an (aggregate) signature τ for C, outputs 1, if the signature is valid, and 0 otherwise.

We require Σ to be *correct* in the sense that for any $k \in \mathbb{N}$:

$$(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Verify}') \text{ is correct } \wedge\ [\mathsf{Verify}(C_1, \tau_1) = 1 \wedge \mathsf{Verify}(C_2, \tau_2) = 1 \Rightarrow \mathsf{Ver}(C_1, C_2, \tau_1, \tau_2) = 1],$$

where

$$\mathsf{Verify}'(C, \sigma) = \begin{cases} \mathsf{Verify}(C, \sigma), & |C| = 1 \\ 0, & \text{else} \end{cases}$$

Security Notion for Aggregate Signatures. Following [BGLS03], we define a security notion for aggregate signature schemes in the aggregate chosen-key model, dubbed EUF-agg-CMA security, where the security notion and experiment is defined in Definition 18, and briefly described here:

- The challenger generates a pair of keys $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ and gives the public key pk to the adversary.
- The adversary F may (adaptively) issue signature queries M_i to a signature oracle, which responds with $\sigma_i \leftarrow \mathsf{Sig}(sk, M_i)$.
- Finally, F outputs a multiset of public key and message pairs C^* and an (aggregate) signature τ^*.

The adversary *wins* the experiment iff there is a message M^* such that $c^* = (pk, M^*)$ is in C^*, $\mathsf{Verify}(C^*, \tau^*) = 1$, and M^* has never been submitted to the signature oracle.

Definition 18 (EUF-agg-CMA)**.** An aggregate signature scheme Σ is *existentially unforgeable in the aggregate chosen-key model under chosen message attacks* (EUF-agg-CMA-secure) if

$$\mathsf{Adv}^{\mathsf{euf\text{-}agg\text{-}cma}}_{\Sigma,F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}agg\text{-}cma}}_{\Sigma,F}(k) = 1\right]$$

is negligible for any PPT adversary F. The experiment $\mathsf{Exp}^{\mathsf{euf\text{-}agg\text{-}cma}}_{\Sigma,F}(k)$ is defined in Figure 4.1.

Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}agg\text{-}cma}}_{\Sigma,F}(k)$
  $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
  $(C^*, \tau^*) \leftarrow F^{\mathsf{Sig}(sk,\cdot)}(pk)$
  if $\mathsf{Verify}(C^*, \tau^*) = 1$
    and $\exists\, c^* =: (pk, M^*) \in C^*$
    and F has not queried $\mathsf{Sig}(sk, M^*)$
    return 1
  else
    return 0

Figure 4.1: EUF-agg-CMA experiment for aggregate signature schemes.

Cover-Free Families. For our construction of a fault-tolerant aggregate signature scheme in section 4.4 and section 4.5 we employ a d-cover-free family [Für96], which allows us to detect up to d invalid individual signatures in our aggregate signature.

Definition 19 (d-Cover-Free Family, [Für96])**.** A *d-cover-free family* $\mathcal{F} = (\mathcal{S}, \mathcal{B})$ (denoted by d-CFF) consists of a set $\mathcal{S}$ of m elements, the *universe*, and a set $\mathcal{B}$ of n subsets of $\mathcal{S}$, where $d < m < n$, such that: For any d subsets $B_{i_1}, \ldots, B_{i_d} \in \mathcal{B}$ and all $B \in \mathcal{B} \setminus \{B_{i_1}, \ldots, B_{i_d}\}$, it holds that

$$|B \setminus \bigcup_{k=1}^{d} B_{i_k}| \geq 1.$$

In other words, it is not possible to cover a single subset with at most d different subsets. To get a better representation of a d-CFF and to simplify the handling of it, we will use a matrix in the following way:

Definition 20 (Incidence Matrix)**.** For a d-CFF $\mathcal{F} = (\mathcal{S}, \mathcal{B})$, where the elements of $\mathcal{S}$ and $\mathcal{B}$ have a well-defined order, such that we can write $\mathcal{S} = \{s_1, \ldots, s_m\}$, $\mathcal{B} = \{B_1, \ldots, B_n\}$, we define its *incidence matrix* $\mathcal{M} \in \{0,1\}^{m \times n}$ as follows:

$$\mathcal{M}[i,j] = \begin{cases} 1, & \text{if } s_i \in B_j, \\ 0, & \text{otherwise.} \end{cases}$$

The *i-th row* of $\mathcal{M}$ is denoted by $\mathcal{M}_i \in \{0,1\}^n$, for $i \in \{1,\ldots,m\}$.

So, $s_i \in \mathcal{S}$ corresponds to row i and $B_j \in \mathcal{B}$ corresponds to column j, i.e. $\text{rows}(\mathcal{M}) = m$ (number of rows) and $\text{cols}(\mathcal{M}) = n$ (number of columns).

4.3 Fault-Tolerant Aggregate Signatures

In this section we explain the concept of *fault-tolerance* and mention some useful notations, to simplify the syntax of a fault-tolerant aggregate signature scheme.

Claims and Claim Sequences. As a notational convenience, we introduce the concept of claims.

Definition 21 (Claim)**.** A *claim* c is a pair (pk, M) of a public key and a message M, conveying the meaning that the owner of pk has authenticated the message M.

In this sense, a signature σ for M that is valid under pk is a proof for the claim c. This definition simplifies the representation of our algorithms.

The signature scheme we introduce in section 4.4 requires an order among the claims. Since the actual order is arbitrary, it must be supported by the aggregation and verification algorithms. We therefore define fault-tolerant signature schemes based on sequences of claims, instead of multisets of claims. We now give a formal definition and explain in more detail afterwards.

Definition 22 (Claim Sequence, claim placeholder)**.** A *claim sequence* C is a tuple of claims c and *claim placeholders* $\perp$. The multiset of elements of a claim sequence C excluding $\perp$ is denoted by $\mathsf{elem}\,(C)$.

Example 1. $C = (c_1, c_2, \perp, c_4)$ is a claim sequence of length 4 and contains the claims c_1, c_2, c_4 on position $1, 2$ and 4 and one claim placeholder $\perp$ on position 3. The multiset of elements of C is $\mathsf{elem}\,(C) = \{c_1, c_2, c_4\}$.

We make use of claim placeholders $\perp$, since in the general aggregation setting we have to deal with 'incomplete' claim sequences, i.e. a claim sequence does not necessarily contain a claim at position j.

When aggregating the signatures of two such incomplete claim sequences C_1, C_2, the claim sequences will be merged, we write $C_1 \sqcup C_2$, meaning that claim placeholders in C_1 are replaced by actual claims from C_2, for each position j where $C_1[j] = \perp$ and $C_2[j] \neq \perp$, and vice versa. (This merging operation replaces the multiset union used by general aggregate signature schemes.)

More precisely, when we want to aggregate an individual signature σ for a claim c into an aggregate signature τ' corresponding to claim sequence C', we have to assign a unique 'position' j to c, such that $C'[j] = \perp$, for $j \in \{1,\ldots,|C'|\}$ or $j := |C'| + 1$, to obtain a new aggregate signature τ with a new corresponding claim sequence C, where c is included on position j, i.e. $C[j] = c$. If one wishes to verify τ, one must call Verify with the claim sequence C. Therefore, two aggregate signatures τ_1, τ_2 for two claim sequences C_1, C_2 can not be aggregated if $C_1[j] \neq \perp$ and $C_2[j] \neq \perp$ for some j.

Thus, our scheme does not support *fully* flexible, arbitrary aggregation, i.e. without regarding any order. However, if the signers agree in advance on the positions j of their claims, they can aggregate all their signatures into a single combined signature τ. This precondition can easily be satisfied in many applications. In wireless sensor networks for example, one only has to configure each sensor to use a different predefined position j. Moreover, it is always possible to use our

scheme as a sequential aggregate signature scheme (for more details see [LMRS04b]), since the position j of a claim needs only be determined when it is first aggregated. Hence, our scheme fits for all applications where sequential aggregate signatures are sufficient, too, such as secure logging [MT08].

More formally, we define the mergeability of claim sequences as follows:

Definition 23 (mergeable, exclusively mergeable)**.** Two claim sequences C_1, C_2 are *mergeable* if for all $i \in \{1, \ldots, \min(|C_1|, |C_2|)\}$ it holds that $C_1[i] = \perp$ or $C_2[i] = \perp$ or $C_1[i] = C_2[i]$. C_1, C_2 are called *exclusively mergeable*, if for all such i it holds that $C_1[i] = \perp$ or $C_2[i] = \perp$.

Definition 24 (merged claim sequence, empty signature)**.** Let C_1 and C_2 be two mergeable claim sequences of length k and l, respectively. Without loss of generality, assume $k \leq l$. Then the *merged claim sequence* $C_1 \sqcup C_2$ is $(c_1, \ldots, c_l)$, where

$$c_i := \begin{cases} C_1[i], & \text{if } C_2[i] = \perp,\ C_2[i] = C_1[i] \text{ or } i > k, \\ C_2[i], & \text{otherwise.} \end{cases}$$

The *empty signature* λ is a signature valid for exactly the claim sequences containing only $\perp$ and the empty claim sequence, defined as $\langle\rangle$.

In particular, two exclusively mergeable sequences are mergeable.

Example 2. For example, let c_1, c_2, c_3 be distinct claims. Define $C_1 := (\perp, c_2, c_3)$, $C_2 := (c_1, \perp, \perp)$, $C_2' := (c_1, c_2, \perp)$, $C_2'' := (c_1, c_3, c_2)$. Then,

- C_1, C_2 are exclusively mergeable, $C_1 \sqcup C_2 = (c_1, c_2, c_3)$, because for all $i \in \{1, 2, 3\}$ either $C_1[i] = \perp$ or $C_2[i] = \perp$.
- C_1, C_2' are mergeable, but not exclusively mergeable, $C_1 \sqcup C_2' = (c_1, c_2, c_3)$, because $C_1[2] = C_2'[2] \neq \perp$.
- C_1, C_2'' are not mergeable, since e.g. $\perp \neq C_1[2] \neq C_2''[2] \neq \perp$.

For technical reasons, as already mentioned we only allow exclusivley mergeable claim sequences C_1 and C_2 as input of our aggregation algorithm (i.e. there is no position where C_1 and C_2 both contain a claim, even if these claims are identical). As a consequence, if a signature τ is aggregated into two different aggregate signatures τ_1, τ_2 using the same position j, τ_1 and τ_2 can not be aggregated. Note, however, that this does not exclude the possibility to aggregate τ into τ_1 and τ_2 at different positions.

Definition 25 (Subsequences)**.** Let $C = (c_1, \ldots, c_n), n \in \mathbb{N}$, be a tuple and $b \in \{0,1\}^n$ be a bit sequence specifying a selection of indices. Then $C[b]$ is the subsequence of C of length n containing exactly the elements c_j where $b[j] = 1$, replacing all other claims by $\perp$.

Example 3. In particular, if $\mathcal{M}$ is an incidence matrix of a cover-free family $\mathcal{F} = (\mathcal{S}, \mathcal{B})$, then $C[\mathcal{M}_i]$ is the subsequence containing all c_j, where $\mathcal{M}[i, j] = 1$, for $j = 1, \ldots, |\mathcal{B}|$, and $\perp$ at all other positions.

Syntax of Fault-Tolerant Aggregate Signature Schemes. We are now prepared to define fault-tolerant aggregate signature schemes. The main difference of such a scheme compared to an ordinary aggregate signature scheme is that its verification algorithm does not only output a boolean value 1 or 0 that identify if either all claims are valid or at least one claim is invalid, but it outputs a multiset of valid claims.

Thus, the ouput of the verification algorithm gives (some) information on which claims in C are valid. If the signature, for example, include more errors than the scheme can cope with, Verify may output just a subset of the valid claims. Other claims may be clearly false or just

not certainly true. (The verification algorithm ought to be conservative and reject a claim in case of uncertainty.)

The aggregation algorithm is called with two claim sequences, hence, before aggregating, a single claim c must be converted to a claim sequence $C := (\bot, \ldots, \bot, c)$ by assigning a position to c.

Definition 26. An *aggregate signature scheme with list verification*[1] $\Sigma = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Agg}, \mathsf{Verify})$ and message space $\mathcal{M}_k$ consists of four PPT algorithms as follows:

Key Generation. $\mathsf{Gen}(1^k)$, on input 1^k, outputs a key pair (pk, sk).

Sign. $\mathsf{Sig}(sk, M)$, on input a secret key sk and a message M, outputs a signature σ.

Aggregation. $\mathsf{Agg}(C_1, C_2, \tau_1, \tau_2)$, on input two exclusively mergeable claim sequences C_1 and C_2 and corresponding (aggregate) signatures τ_1 and τ_2, outputs (C, τ), where τ is an aggregate signature, certifying the validity of the claim sequence $C := C_1 \sqcup C_2$.

List Verification $\mathsf{Verify}(C, \tau)$, on input a claim sequence C and an (aggregate) signature τ for C, outputs a multiset of claims $\mathcal{C}_{\text{valid}} \subseteq \mathsf{elem}\,(C)$ specifying the valid claims in C for τ.

Note that $\mathcal{C}_{\text{valid}}$ may be a proper subset of $\mathsf{elem}\,(C)$, or even empty, if none of the claims can be derived from τ (for certain). Again, here, C may contain $\bot$ as a claim placeholder.

For *correctness* we require the following paragraphs.

Regular Signatures. Informally, a signature is regular if it is created by running the algorithms of Σ. More formally, consider the following definition.

Definition 27 (Regular Signatures)**.** Let C be a claim sequence and τ be an (aggregate) signature. We recursively define what it means for τ to be *regular for* C:

- If (pk, sk) is in the image of $\mathsf{Gen}(1^k)$ and $C = ((pk, M))$ for a message M, and if τ is in the image of $\mathsf{Sig}(sk, M)$, then τ is said to be regular for C and for any claim sequence obtained by prepending any number of $\bot$ symbols to C.
- If τ_1 is regular for a claim sequence C_1, τ_2 is regular for a claim sequence C_2, and C_1, C_2 are exclusively mergeable, then τ is regular for $C_1 \sqcup C_2$ if τ is in the image of $\mathsf{Agg}(C_1, C_2, \tau_1, \tau_2)$.
- The empty signature λ is regular for the claim sequences containing only $\bot$ and the empty claim sequence $\langle\rangle$.

If an (aggregate) signature τ is not regular for a claim sequence C, it is called *irregular* for C.

Fault-Tolerance. Informally, an aggregate signature scheme with list verification is fault-tolerant, if it still outputs all valid claims, even if there are some faulty claims. More formally, consider the following definitions.

Definition 28 (Tolerance against d errors)**.** Let $S = \{(c_1, \sigma_1), \ldots, (c_n, \sigma_n)\}$ be a multiset of claim and signature pairs, which is partitioned into two multisets S_{reg} and S_{irreg}, containing the pairs for which σ_i is regular for $C_i = (c_i)$ and irregular for C_i, respectively.[2]

- Then the multiset S *contains* d *errors*, if $|S_{\text{irreg}}| = d$.

[1] The name 'list verification' is chosen to indicate the changes in syntax, in particular that the verification algorithm outputs a multiset (list) instead of just 1 or 0.

[2] While there may be schemes with valid signatures which are not regularly generated, like in the usual correctness properties, our guarantees do only concern regular signatures.

- An aggregate signature scheme Σ with list verification is *tolerant against d errors*, if for any such multiset S containing at most d errors, for any aggregate signature τ that was aggregated from the signatures in S (in arbitrary order) and the corresponding claim sequence C, which may additionally contain any number of claim placeholders $\perp$, we have

$$R \subseteq \mathcal{C}_{\text{valid}} := \mathsf{Verify}(C, \tau),$$

  where R is the multiset of all claims c_i (i.e. the first component of the pairs) in S_{reg}.

Remark 3. In other words, Verify outputs at least all claims of regular signatures. Intuitively, one would expect $R = \mathcal{C}_{\text{valid}}$. However, this is not achievable in general, as the aggregation of multiple irregular signatures may contain a new valid claim c_i corresponding to an irregular signature σ_i. This does not contradict security, as crafting such irregular signatures may be hard if one does not know σ_i. For an example see Example 6 in section 4.5.

Definition 29 (Fault-Tolerance)**.** For $d \in \mathbb{N}$ we define:

- A *d-fault-tolerant aggregate signature scheme* is an aggregate signature scheme with list verification that is tolerant against d errors.
- A *fault-tolerant aggregate signature scheme* is a scheme that is d-fault-tolerant for some $d > 0$.

From now on, we denote a fault-tolerant aggregate signature scheme (with list verification) as Σ_{ft}.

Correctness. Observe that 0-fault-tolerance means that if S contains only regularly created signatures, then Verify must output all claims in S (or C, respectively). This is analogous to the common definition of correctness for aggregate signature schemes.

Definition 30 (Correctness)**.** We say an aggregate signature scheme with list verification is *correct*, if it is tolerant against 0 errors.

Errors During Aggregation. Our definitions above assume that aggregation is always done correctly. This is a necessary assumption, since it is impossible to give guarantees for arbitrary errors that happen during aggregation. Consider for example a faulty aggregation algorithm that ignores its input and just outputs a random string. It is an interesting open question to find a fault-tolerant signature scheme that can tolerate certain types of aggregation errors, too.

Size of Fault-Tolerant Aggregate Signatures. A typical attribute of an aggregate signature scheme is that the length of an aggregate signature is (almost) the same as that of an individual signature [HSW13]. Furthermore, the number of signatures that can be aggregated into a single signature should be unbounded.

We show that these goals are mutually exclusive for a fault-tolerant aggregate signature schemes if one wishes to maintain a constant $d \geq 1$.

Theorem 8. Let $n, d \in \mathbb{N}$, and $\Sigma_{ft} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Agg}, \mathsf{Verify})$ be a d-fault-tolerant signature scheme. Assume that $\mathcal{C}_{\text{valid}} = R$ for all claim sequences C and corresponding aggregate signatures τ constructed from an arbitrary multiset $S = \{(c_1, \sigma_1), \ldots, (c_n, \sigma_n)\}$ of n claim signature pairs and containing at most d errors, and where R is the multiset of all claims c_i accompanied by a regular signature σ_i in S. Then we have

$$|\tau| \in \mathbf{\Omega}(\log_2 n)$$

as a function of n, where d is considered constant, and $|\tau|$ is the length of the aggregate signature τ in bits.

The proof can be found in [HKKKR16] and will be discussed in another thesis.

Definition 31 (Compression Ratio)**.** Denote by $\mathsf{size}(\sigma)$ the size of a signature σ. Let C be a claim sequence of length n, and τ an aggregate signature of maximum size[3] which is regular for C. We say that an aggregate signature scheme has *compression ratio* $\rho(n)$ if

$$\frac{n}{\mathsf{size}(\tau)} \in \Theta(\rho(n)).$$

Note that if $\mathsf{size}(\tau)$ is upper bounded by a constant, then the compression ratio is $\rho(n) = n$, which is optimal for common aggregate signature schemes, but as already explained above this is not possible for fault-tolerant aggregate signatures.

Security Notion for Fault-Tolerant Aggregate Signatures. The security notion and experiment for fault-tolerant aggregate signatures schemes (with list verification), dubbed EUF-ft-agg-CMA, is a direct adaption of the EUF-agg-CMA-security notion and experiment for aggregate signature schemes from section 4.2 with a slightly modification, defined in Definition 32, and briefly described here:

- The challenger generates a pair of keys $(pk, sk) \leftarrow \mathsf{Gen}(1^k)$ and gives the public key pk to the adversary.
- The adversary F may (adaptively) issue signature queries M_i to the signature oracle, which responds with $\sigma_i \leftarrow \mathsf{Sig}(sk, M_i)$.
- Finally, F outputs a claim sequence C^* and an (aggregate) signature τ^*.

The adversary *wins* the experiment iff there is a message M^* such that $c^* = (pk, M^*) \in \mathsf{Verify}(C^*, \tau^*)$, and M^* has never been submitted to the signature oracle.

Definition 32 (EUF-ft-agg-CMA)**.** A fault-tolerant aggregate signature scheme (with list verification) Σ_{ft} is *existentially unforgeable in the aggregate chosen-key model under chosen message attacks* (EUF-ft-agg-CMA-secure) iff

$$\mathsf{Adv}^{\mathsf{euf\text{-}ft\text{-}agg\text{-}cma}}_{\Sigma_{ft},F}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{euf\text{-}ft\text{-}agg\text{-}cma}}_{\Sigma_{ft},F}(k) = 1\right]$$

is negligible for any PPT adversary F. The experiment $\mathsf{Exp}^{\mathsf{euf\text{-}ft\text{-}agg\text{-}cma}}_{\Sigma_{ft},F}(k)$ is defined in Figure 4.2.

Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}ft\text{-}agg\text{-}cma}}_{\Sigma_{ft},F}(k)$
$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$
$(C^*, \tau^*) \leftarrow F^{\mathsf{Sig}(sk,\cdot)}(pk)$
if $\mathsf{Verify}(C^*, \tau^*) =: \mathcal{C}_{\text{valid}}$
  and $\exists\, c^* = (pk, M^*) \in \mathcal{C}_{\text{valid}}$
  and F has not queried $\mathsf{Sig}(sk, M^*)$
  return 1
else
  return 0

Figure 4.2: EUF-ft-agg-CMA experiment for fault-tolerant aggregate signature schemes (with list verification).

[3]The size of an aggregated signature might depend on the aggregation order.

4.4 Generic Construction

In this section we give a generic construction of a fault-tolerant aggregate signature scheme (with list verification). We emphasize here that this part will also be discussed in more detail in another thesis, with respect to the correctness and security proof.

Our Construction. The construction is based on an arbitrary aggregate signature scheme Σ', which is used as a black box, and a cover-free family $\mathcal{F} = (\mathcal{S}, \mathcal{B})$, where $|\mathcal{S}| = m$, and $|\mathcal{B}| = n$, for $m \ll n$. Our scheme inherits its security from Σ', and can tolerate d faults if it uses a d-cover-free family.

Let $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Agg}', \mathsf{Verify}')$ be an ordinary aggregate signature scheme according to Definition 17. Moreover, let $\mathcal{M}$ be the incidence matrix of a d-cover-free family $\mathcal{F} = (\mathcal{S}, \mathcal{B})$, as defined in Definition 20 (In section 4.5 we provide a concrete instantiation).

We first show a bounded construction and in section 4.4.1 an unbounded extension. In the bounded construction, the maximum number of signatures that can be aggregated is $\mathrm{cols}(\mathcal{M}) = |\mathcal{B}| = n$.

In our scheme, signatures for just one claim are simply signatures of the underlying scheme Σ', whereas aggregate signatures are short vectors of length $\mathrm{rows}(\mathcal{M}) = |\mathcal{S}| = m$ of signatures of Σ'. We identify each element of the universe $\mathcal{S}$ with a position in this vector, and each subset $B \in \mathcal{B}$ with an individual signature of the underlying scheme Σ'.

We require that the underlying scheme Σ' supports claim sequences and claim placeholders as an input to Agg' and Verify', contrary to just multisets, as in the definition of standard aggregate signature schemes (Definition 17).

Moreover, we assume that Σ' supports the empty signature λ as an input to Agg' and Verify'. However, these are not essential restrictions, as for instance any standard aggregate scheme may be easily adapted to a scheme of the modified syntax, by ignoring any order and claim placeholders, i.e. applying $\mathsf{elem}\,(\cdot)$ on the claim sequences before they are passed to the Agg' and Verify' algorithm.

More formally, let $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Agg}', \mathsf{Verify}')$ be an aggregate signature scheme, with message space $\mathcal{M}_k$. Let $\mathcal{F} = (\mathcal{S}, \mathcal{B})$ be a d-cover-free family, with incidence matrix $\mathcal{M}$, where $|\mathcal{S}| = m = \mathrm{rows}(\mathcal{M})$, and $|\mathcal{B}| = n = \mathrm{cols}(\mathcal{M})$. Our fault-tolerant aggregate signature scheme (with list verification) $\Sigma_{ft} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Agg}, \mathsf{Verify})$ consists of the following four algorithms (see also Figure 4.2):

Key Generation. $\mathsf{Gen}(1^k)$, on input 1^k, runs $(pk', sk') \leftarrow \mathsf{Gen}'$, and outputs a key pair $(pk, sk) := (pk', sk')$.

Sign. $\mathsf{Sig}(sk, M)$, on input a secret key sk and a message M, runs $\sigma' \leftarrow \mathsf{Sig}'(sk, M)$, and outputs a signature $\sigma := \sigma'$.

Aggregation. $\mathsf{Agg}(C_1, C_2, \tau_1, \tau_2)$, on input two exclusively mergeable claim sequences C_1 and C_2, and corresponding (aggregate) signatures τ_1 and τ_2, proceeds as follows:

1. If one or both of the claim sequences C_l ($l \in \{1, 2\}$) contains only one (proper) claim c, i.e. τ_l is an individual signature, then σ_l is initialized as τ_l, the corresponding signature given to Agg'. Then τ_l is expanded to a vector, by setting

$$\tau_l[i] := \begin{cases} \sigma_l, & \text{if } \mathcal{M}[i, j] = 1, \\ \lambda, & \text{otherwise,} \end{cases} \qquad \text{for } i = 1, \ldots, m,$$

   where j is the index of c in the claim sequence.

2. Then the (aggregate) signatures τ_1, τ_2, which are both vectors now, are aggregated component-wise, i.e.

$$\tau[i] := \mathsf{Agg}'(C_1[\mathcal{M}_i], C_2[\mathcal{M}_i], \tau_1[i], \tau_2[i]).$$

$\mathsf{Gen}(1^k)$

$(pk', sk') \leftarrow \mathsf{Gen}'(1^k)$
$pk := pk'$
$sk := sk'$
return (pk, sk)

$\mathsf{Sig}(sk, M)$

$\sigma' \leftarrow \mathsf{Sig}'(sk, M)$
$\sigma := \sigma'$
return σ

$\mathsf{Agg}(C_1, C_2, \tau_1, \tau_2)$

for $l := 1$ **to** 2 **do**
  if $|\mathsf{elem}\,(C_l)\,| = 1$ **then**
    assign position j
    $\sigma_l := \tau_l$
    for $i := 1$ **to** m **do**
      if $\mathcal{M}[i, j] = 1$ **then**
        $\tau_l[i] := \sigma_l$
      else $\tau_l[i] := \lambda$
    //end for
  //end if
//end for
for $i := 1$ **to** m **do**
  $\tau[i] := \mathsf{Agg}'(C_1[\mathcal{M}_i], C_2[\mathcal{M}_i], \tau_1[i], \tau_2[i])$
$C := C_1 \sqcup C_2$
return (C, τ)

$\mathsf{Verify}(C, \tau)$

$\mathcal{C}_{\text{valid}} := \emptyset$
for $i := 1$ **to** m **do**
  $b_i := \mathsf{Verify}'(C[\mathcal{M}_i], \tau[i])$
  if $b_i = 1$ **then**
    $\mathcal{C}_{\text{valid}} := \mathcal{C}_{\text{valid}} \cup \mathsf{elem}\,(C[\mathcal{M}_i])$
  //end if
//end for
return $\mathcal{C}_{\text{valid}}$

Figure 4.3: Our EUF-ft-agg-CMA-secure d-fault-tolerant aggregate signature scheme (with list verification).

Finally, Agg outputs (C, τ), where $\tau = \begin{pmatrix} \tau[1] \\ \vdots \\ \tau[m] \end{pmatrix}$ is an aggregate signature, certifying the validity of the claim sequence $C := C_1 \sqcup C_2$.

List Verification. $\mathsf{Verify}(C, \tau)$, on input a claim sequence C and an (aggregate) signature τ for C, computes for each component $\tau[i]$ of τ

$$b_i \coloneqq \mathsf{Verify}'(C[\mathcal{M}_i], \tau[i]), \text{ for } i = 1, \ldots, m,$$

and the multiset of valid claims

$$\mathcal{C}_{\text{valid}} \coloneqq \bigcup_{i \in \{1,\ldots,m\}, b_i = 1} \mathsf{elem}\,(C[\mathcal{M}_i])$$

and outputs $\mathcal{C}_{\text{valid}}$.

The next theorem considers the correctness and security of our scheme and is further discussed in another thesis.

Theorem 9. If Σ' is an EUF-agg-CMA-secure aggregate signature scheme, then the scheme Σ_{ft} defined above, based on a d-CFF, is an EUF-ft-agg-CMA-secure d-fault-tolerant aggregate signature scheme (with list verification), and is correct.

The proof can be found in [HKKKR16].

Illustration. We want to illustrate our construction here for a better understanding. Let $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Agg}', \mathsf{Verify}')$ be an ordinary aggregate signature scheme (for instance BGLS). We set our parameters in the following way:

$$m = 4, n = 6, d = 1.$$

This means our aggregate signature τ will be a vector of 4 elements

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix}$$

and we can aggregate up to 6 individual signatures $\sigma_1, \ldots, \sigma_6$ with

$$\sigma_j \leftarrow \mathsf{Sig}'(sk, M_j),\ j = 1, \ldots, 6.$$

Thus, we have to choose a 4×6 incidence matrix $\mathcal{M}$ that can tolerate $d = 1$ faulty signature, e.g.:

$$\mathcal{M} := (m_{i,j}) := \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

According to this, the 1-entries in column j indicate the positions, where we embed our individual signatures σ_j, i.e.

$$\begin{pmatrix} \sigma_1 & 0 & 0 & \sigma_4 & 0 & \sigma_6 \\ \sigma_1 & \sigma_2 & 0 & 0 & \sigma_5 & 0 \\ 0 & \sigma_2 & \sigma_3 & \sigma_4 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & \sigma_5 & \sigma_6 \end{pmatrix}.$$

Now, we use the underlying aggregate signature scheme Σ' and aggregate the signatures in one row, which results in

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} = \begin{pmatrix} \mathsf{Agg}'(\sigma_1, \sigma_4, \sigma_6) \\ \mathsf{Agg}'(\sigma_1, \sigma_2, \sigma_5) \\ \mathsf{Agg}'(\sigma_2, \sigma_3, \sigma_4) \\ \mathsf{Agg}'(\sigma_3, \sigma_5, \sigma_6) \end{pmatrix} \hat{=} \begin{pmatrix} \sigma_1 & & & \sigma_4 & & \sigma_6 \\ \sigma_1 & \sigma_2 & & & \sigma_5 & \\ & \sigma_2 & \sigma_3 & \sigma_4 & & \\ & & \sigma_3 & & \sigma_5 & \sigma_6 \end{pmatrix}.$$

If exactly one signature σ_j is invalid, all τ_i are faulty where $m_{i,j} = 1$.

Example 4. For a concrete example, suppose σ_2 is invalid. Then τ_2 and τ_3 will be invalid, whereas τ_1 and τ_4 are valid. We see that σ_1, σ_4 and σ_6 occur in τ_1, and σ_3 and σ_5 occur in τ_4, and so we may be sure that the corresponding messages were signed correctly under the corresponding public keys. Hence, Verify will output

$$\mathcal{C}_{\text{valid}} = \mathsf{elem}\,(C[\mathcal{M}_1]) \cup \mathsf{elem}\,(C[\mathcal{M}_4]) = \{c_1, c_3, c_4, c_5, c_6\}$$

Unfortunately, this is not possible if two or more faulty signatures are aggregated. Lets assume that σ_1 and σ_2 are invalid. In this case, τ_1, τ_2 and τ_3 become invalid and τ_4 is the only valid aggregate signature. We could still derive the validity of $\sigma_3, \sigma_5, \sigma_6$, because τ_4 is valid. However, the validity of σ_4 can no longer be verified, since σ_4 is not part of τ_4.

Compression Ratio. Let $C = (c_1, \ldots, c_n), n \in \mathbb{N}$, be a claim sequence of length n, and τ be an aggregate signature regular for C. We assume in the following that the length of all signatures of the underlying scheme Σ' is bounded by a constant in n (but poly in k) *const* and is at least 1. Then the compression ratio of our scheme is

$$\rho(n) = \frac{n}{\text{rows}(\mathcal{M})},$$

since

$$\frac{n}{\mathsf{size}(\tau)} \leq \frac{n}{\text{rows}(\mathcal{M}) \cdot const} \in \mathbf{O}(\rho(n)) \quad \text{and} \quad \frac{n}{\mathsf{size}(\tau)} \geq \frac{n}{\text{rows}(\mathcal{M})} \in \Omega(\rho(n)). \tag{4.1}$$

Clearly, the compression ratio $\rho(n)$ of our scheme is less than 1 if $n < \text{rows}(\mathcal{M})$, and the resulting aggregate signature is larger than the sum of the individual signature sizes when only few signatures have been aggregated so far. Our scheme can be easily adapted to fix this behavior, by simply storing all individual signatures instead of immediately aggregating them, until $n = \text{rows}(\mathcal{M})$. When the $n+1$-st signature is added, the individual signatures are aggregated using the aggregation algorithm defined above. When further signatures are added, the size of the aggregate signature remains bounded by $\text{rows}(\mathcal{M}) \cdot const$.

4.4.1 Unbounded Aggregation

In order to achieve unbounded aggregation, we do not need just one cover-free family, but a sequence of cover-free families increasing in size, such that we can switch to the next larger one, as soon as we exceed the capacity for the number of aggregatable signatures. This sequence needs to exhibit a monotonicity property, in order to work with our scheme, which we define next.

Definition 33 (Monotone Family)**.** We consider a family $(\mathcal{M}^{(\lambda)})_\lambda$ of incidence matrices of corresponding d-cover-free families $(\mathcal{F}_\lambda)_\lambda := (\mathcal{S}_\lambda, \mathcal{B}_\lambda)_\lambda$, where $\text{rows}(\lambda)$ denotes the number of rows and $\text{cols}(\lambda)$ denotes the number of columns of $\mathcal{M}^{(\lambda)}$. $(\mathcal{M}^{(\lambda)})_\lambda$ is a *monotone family* of incidence matrices of $(\mathcal{F}_\lambda)_\lambda$, if for $\lambda \geq 1$,

$$\mathcal{S}_\lambda \subseteq \mathcal{S}_{\lambda+1},$$

$$\mathcal{B}_\lambda \subseteq \mathcal{B}_{\lambda+1},$$

s.t.

$$\mathcal{S}_{\lambda+1} = \{s_1, \ldots, s_{\text{rows}(\lambda)}, s_{\text{rows}(\lambda)+1}, \ldots, s_{\text{rows}(\lambda+1)}\}$$

and

$$\mathcal{B}_{\lambda+1} = \{B_1, \ldots, B_{\text{cols}(\lambda)}, B_{cols(\lambda)+1}, \ldots, B_{\text{cols}(\lambda+1)}\},$$

where $\mathcal{S}_\lambda = \{s_1, \ldots, s_{\text{rows}(\lambda)}\}$ and $\mathcal{B}_\lambda = \{B_1, \ldots, B_{\text{cols}(\lambda)}\}$.

Note that Definition 33 implies that

$$\mathcal{M}^{(\lambda+1)} = \begin{pmatrix} \mathcal{M}^{(\lambda)} & \mathbf{A} \\ \mathbf{0} & \mathbf{B} \end{pmatrix}$$

where for $i = 1, \ldots, \text{rows}(\lambda), j = \text{cols}(\lambda) + 1, \ldots, \text{cols}(\lambda + 1)$

$$\mathbf{A}[i,j] = \begin{cases} 1, & \text{if } s_i \in B_j, \\ 0, & \text{otherwise} \end{cases}$$

and for $i = \text{rows}(\lambda), \ldots, \text{rows}(\lambda+1), j = \text{cols}(\lambda)+1, \ldots, \text{cols}(\lambda+1)$

$$\mathbf{B}[i,j] = \begin{cases} 1, & \text{if } s_i \in B_j, \\ 0, & \text{otherwise.} \end{cases}$$

So, each $\mathcal{M}^{(\lambda)}$ contains all previous $\mathcal{M}^{(1)}, \ldots, \mathcal{M}^{(\lambda-1)}$.

Now, we are able to achieve unbounded aggregation, i.e. our construction is able to aggregate an arbitrary number of signatures, by replacing the fixed incidence matrix $\mathcal{M}$ of a d-CFF in our construction with a monotone family of incidence matrices $(\mathcal{M}^{(\lambda)})_\lambda$.

For this, a run of our aggregation algorithm Agg on inputs C_1, C_2, τ_1, τ_2 first has to determine the smallest λ, such that

$$\text{cols}(\lambda) \geq \max(|C_1|, |C_2|)$$

and then proceeds with the corresponding incidence matrix $\mathcal{M}^{(\lambda)}$.

Analogously, our verification algorithm Verify on inputs C, τ first determines the smallest λ such that

$$\text{cols}(\lambda) \geq |C|$$

.

Compression Ratio. The compression ratio of our unbounded scheme is $\rho(n) = n/\text{rows}(\lambda)$, where λ is the minimum index such that $\text{cols}(\lambda) \geq n$.

4.4.2 Selective Verification

Let τ be a regular (aggregate) signature with corresponding claim sequence $C = (c_1, \ldots, c_n), n \in \mathbb{N}$. Assume we want to know whether a signature for a specific claim c^* was correctly aggregated into τ, but we want to avoid verifying all the claims in C to save verification time, especially if C is large. It is a unique feature of our fault-tolerant aggregate signature scheme that there is an additional algorithm, denoted by $\mathsf{SelectiveVerify}(C, \tau, c^*)$ (see also Figure 4.4), that outputs the number of occurrences of c^* in C that have a valid signature in τ, i.e., the number of occurrences of c^* in $\mathsf{Verify}(C, \tau)$, while being faster than actually calling $\mathsf{Verify}(C, \tau)$.

Let Σ_{ft} be the d-fault-tolerant aggregate signature scheme (with list verification) defined above and Σ' be the underlying aggregate signature scheme. Then $\mathsf{SelectiveVerify}$ works as follows:

- First, it determines the set

  $$J \coloneqq \{j \in \mathbb{N} \colon c_j = c^* \text{ for } c_j \in C\},$$

  i.e. the set of indices j where c^* occurs in C.

- Then it determines the set

  $$I \coloneqq \{i \in \text{rows}(\mathcal{M}) |\ \exists j \in J : M[i,j] = 1\},$$

  i.e. the set of indices of all rows where an individual signature for c^* was aggregated.

- Then, it initializes

  $$\hat{C} := \langle\rangle$$

  and iterates over all $i \in I$, checking if

  $$b_i := \Sigma.\mathsf{Verify}(C[\mathcal{M}_i], \tau[i]) = 1.$$

  If this is the case for an i, it sets

  $$\hat{C} := \hat{C} \sqcup C[\mathcal{M}_i].$$

  As soon as $\hat{C}$ contains $|J|$ occurrences of c^*, $\mathsf{SelectiveVerify}$ skips all remaining $i \in I$.

SelectiveVerify(C, τ, c^*)

$n := |C|$
$J := \emptyset$, $I := \emptyset$, $\mathcal{C}^* := \emptyset$, $\hat{C} := \langle\rangle$
for $j := 1$ **to** n **do**
  if $c_j = c^*$ **then**
    $J := J \cup \{j\}$
for $i := 1$ **to** rows$(\mathcal{M})$
  for $j := 1$ **to** n **do**
    if $j \in J$ **then**
      if $\mathcal{M}[i, j] = 1$ **then**
        $I := I \cup \{i\}$
        break
for $i := 1$ **to** rows$(\mathcal{M})$ **do**
  if $i \in I$ **then**
    if Verify$'(C[\mathcal{M}_i], \tau[i]) = 1$ **then**
      $\hat{C} := \hat{C} \sqcup C[\mathcal{M}_i]$
      if $|\hat{C}| = |J|$ **then**
        break
$\mathcal{C}^* := \{j \in \mathbb{N} : c_j = c^* \text{ for } c_j \in \hat{C}\}$
return $|\mathcal{C}^*|$

Figure 4.4: Additional algorithm SelectiveVerify.

- After the loop is done, SelectiveVerify outputs the number of occurrences of c^* in $\hat{C}$.

Remark 4. Since Verify returns all claims that are contained in a subsequence $C[\mathcal{M}_i]$ with $b_i = 1$, the output of SelectiveVerify is exactly the number of occurrences of c^* in Verify. SelectiveVerify therefore inherits the fault-tolerance and security properties already proven for Verify.

In the best case, SelectiveVerify requires only one call to the underlying verification algorithm Verify$'$. In the worst case, it still only requires

$$|I| \leq \sum_{j \in J} |B_j|$$

calls to Verify$'$, where B_j is the set from the cover-free family corresponding to column j.

Subsignature. Going a little further, it is even possible to create a 'subsignature' for c^* that allows everyone to check that c^* has a valid signature without requiring the complete claim sequence C and the complete signature τ: It is sufficient to give

$$\hat{C} := \bigsqcup_{i \in I} C[\mathcal{M}_i]$$

and the signatures

$$\tau[i] \text{ for } i \in I$$

to the verifier.

4.5 Instantiation with a Cover-Free Family Based on Polynomials over a Finite Field

In this section, we consider a concrete construction of a d-CFF which can be used to instantiate our generic d-fault-tolerant aggregate signature scheme. There are several d-CFF constructions in the literature, for instance, constructions based on concatenated codes [DLVY01; DMR00], polynomials, algebraic-geometric Goppa codes as well as randomized constructions [KRS99]. The following theorem gives a lower bound for the number of rows of the incidence matrix in terms of parameter d and the number of columns.

Theorem 10. For a d-CFF $\mathcal{F} = (\mathcal{S}, \mathcal{B})$, where $|\mathcal{S}| = m$, $|\mathcal{B}| = n$, it holds

$$m \geq const \cdot \frac{d^2}{\log d} \log n$$

for some constant $const \in (0, 1)$.

Proofs can be found in [DVPS14; Für96; Rus94].
In the following construction we use for simplicity only a single incidence matrix. However, by [LVRW06], there is a generic construction to transform an incidence matrix into a monotone family of incidence matrices.

Lemma 11 ([LVRW06])**.** If $\mathcal{F} = (\mathcal{S}, \mathcal{B})$ and $\mathcal{F}' = (\mathcal{S}', \mathcal{B}')$ are d-CFFs, then there exists a d-CFF $\mathcal{F}^* = (\mathcal{S}^*, \mathcal{B}^*)$ with $|\mathcal{S}^*| = |\mathcal{S}| + |\mathcal{S}'|$ and $|\mathcal{B}^*| = |\mathcal{B}| + |\mathcal{B}'|$.

Proof. Suppose $\mathcal{M}$ and $\mathcal{M}'$ are the incidence matrices of d-CFFs $\mathcal{F} = (\mathcal{S}, \mathcal{B})$ and $\mathcal{F}' = (\mathcal{S}', \mathcal{B}')$, respectively. Then

$$\mathcal{M}^* = \begin{pmatrix} \mathcal{M} & \mathbf{0} \\ \mathbf{0} & \mathcal{M}' \end{pmatrix}$$

is an incidence matrix for a d-CFF $\mathcal{F}^* = (\mathcal{S}^*, \mathcal{B}^*)$ with $|\mathcal{S}^*| = |\mathcal{S}| + |\mathcal{S}'|$ and $|\mathcal{B}^*| = |\mathcal{B}| + |\mathcal{B}'|$. □

For our approach we could use a deterministic construction of a d-CFF based on polynomials like [KRS99] did in the following way and for which we propose a generalization to the multivariate case in section 4.5.3.

4.5.1 Construction

For our d-CFF $\mathcal{F} = (\mathcal{S}, \mathcal{B})$ let

$$\mathbb{F}_q = \{x_1, \ldots, x_q\}$$

be a finite field and

$$\mathcal{S} := \mathbb{F}_q^2 = \{(x_i, x_j) \colon i, j = 1, \ldots, q\},$$

and thus

$$|\mathcal{S}| = q^2.$$

For ease of presentation, we assume that q is a prime (as opposed to a prime power), so we may write

$$\mathbb{F}_q = \{0, \ldots, q-1\}.$$

We consider the set

$$\mathbb{F}_q[X]_{\leq l} := \left\{ a_l X^l + \cdots + a_1 X + a_0 \colon a_i \in \mathbb{F}_q,\ i = 0, \ldots, l \right\},$$

of all univariate polynomials $f \in \mathbb{F}_q[X]$ of degree at most l. We have

$$|\mathbb{F}_q[X]_{\leq l}| = q^{l+1}.$$

For every $f \in \mathbb{F}_q[X]_{\leq l}$, we consider the subset

$$B_f := \{(x_1, f(x_1)), \ldots, (x_q, f(x_q))\} \subset \mathcal{S}, \text{ of size } |B_f| = q,$$

consisting of all tuples $(x, y) \in \mathcal{S}$ which lie on the graph of $f \in \mathbb{F}_q[X]_{\leq l}$, i.e. for which $f(x) = y$. From this we obtain

$$\mathcal{B} := \{B_f \colon f \in \mathbb{F}_q[X]_{\leq l}\}, \text{ which is of size}$$
$$|\mathcal{B}| = q^{l+1},$$

since we have $|\mathbb{F}_q[X]_{\leq l}| = q^{l+1}$ different sets B_f in $\mathcal{B}$.
For any distinct $B_f, B_{f_1}, \ldots, B_{f_d} \in \mathcal{B}$ it holds that

$$|B_f \cap B_{f_i}| \leq l, \text{ for } i = 1, \ldots, d$$

since the degree of each polynomial $g_i := f - f_i$ is at most l, i.e.

$$\deg(g_i) \leq l$$

and hence they have at most l zeros. Thus, we have

$$\left| B_f \setminus \bigcup_{i=1}^{d} B_{f_i} \right| \geq q - d \cdot l$$

To achieve a d-CFF with these definitions,

$$q \geq d \cdot l + 1$$

must be satisfied.
Now, we consider the incidence matrix $\mathcal{M}$ of a d-CFF, which consists of $|\mathcal{S}|$ rows and $|\mathcal{B}|$ columns.

$$\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ |\mathcal{S}| \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & \ldots & \ldots & |\mathcal{B}| \end{array} \\ \begin{pmatrix} * & * & \ldots & \ldots & * \\ * & * & \ldots & \ldots & * \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ * & * & \ldots & \ldots & * \end{pmatrix} \end{array} = \mathcal{M}, \text{ where } * \in \{0, 1\}.$$

Each row corresponds to an element of $\mathcal{S}$ and each column to an element of $\mathcal{B}$.

Hence, in the construction above each row corresponds to a tuple $(x, y) \in \mathbb{F}_q^2$ $(= \mathcal{S})$, where we define the order lexicographically, i.e. $(0,0), (0,1), \ldots, (q-1, q-1)$. In the following, let

$$s_i := (x_i, y_i),$$

denote the corresponding tuple for row i, for $i = 0, \ldots, q^2 - 1$, where

$$x_i, y_i \in \{0, \ldots, q-1\}, \text{ s.t } \; i = q \cdot x_i + y_i.$$

Hence,

$$\begin{array}{cccc} s_0 = (0,0), & s_1 = (0,1), & \ldots, & s_{q-1} = (0, q-1), \\ s_q = (1,0), & s_{q+1} = (1,1), & \ldots, & s_{2q-1} = (1, q-1), \\ \vdots & \vdots & \ddots & \\ s_{q^2-q} = (q-1, 0), & s_{q^2-q+1} = (q-1, 1), & \ldots, & s_{q^2-1} = (q-1, q-1). \end{array}$$

Each column of the incidence matrix $\mathcal{M}$ corresponds to a polynomial of degree at most l, where we decide to use lexicographically order starting with constant polynomials and ending with polynomials of degree l, i.e.

$$\begin{array}{rcrrcrrcrcrrcr}
f_0 & := & 0, & f_1 & := & 1, & f_2 & := & 2, & \dots & , & f_{q-1} & := & q-1, \\
f_q & := & X, & f_{q+1} & := & X+1, & f_{q+2} & := & X+2, & \dots & , & f_{2q-1} & := & X+q-1, \\
f_{2q} & := & 2X, & f_{2q+1} & := & 2X+1, & f_{2q+2} & := & 2X+2, & \dots & , & f_{3q-1} & := & 2X+q-1, \\
 & & & & & \vdots & & & & & & & & \\
f_{q^{l+1}-1} & := & (q-1)X^l + (q-1)X^{l-1} + \dots + (q-1)X + q - 1. & & & & & & & & & & &
\end{array}$$

By f_j we denote the corresponding polynomial for column j, for $j = 0, \dots, q^{k+1} - 1$, again starting from 0. Now, the incidence matrix is built as

$$\mathcal{M}[i,j] = \begin{cases} 1, & \text{if } f_j(x_i) = y_i, \\ 0, & \text{otherwise.} \end{cases}$$

Example 5. For $d = 2$ and $l = 2$ we have

$$q \geq d \cdot l + 1 = 2 \cdot 2 + 1 = 5$$

and therefore set $q = 5$ and obtain a 2-CFF with

$$\mathcal{S} = \{(0,0), (0,1), \dots, (4,3), (4,4)\},$$
$$|\mathcal{S}| = 5^2 = 25.$$

We have

$$\mathcal{B} = \{B_{f_0}, \dots, B_{f_{124}}\}, \text{ since } |\mathbb{F}_5[X]_{\leq 2}| = 5^{2+1} = 125,$$

where

$$B_{f_j} = \{(0, f_j(0)), (1, f_j(1)), \dots, (4, f_j(4))\},$$
$$|B_{f_j}| = 5, \quad j = 0, \dots, 5^3 - 1$$

and

$$\begin{array}{rcrrcrrcrcrrcr}
f_0 & := & 0, & f_1 & := & 1, & \dots, & f_4 & := & 4, \\
f_5 & := & X, & f_6 & := & X+1, & \dots, & f_9 & := & X+4, \\
 & & & & & & \vdots & & & \\
f_{120} & := & 4X^2 + 4X, & & & & \dots, & f_{124} & := & 4X^2 + 4X + 4.
\end{array}$$

Thus, we obtain our incidence matrix $\mathcal{M}$ as follows:

$$\begin{array}{c} \\ (0,0) \\ (0,1) \\ \vdots \\ (4,4) \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & \dots & X & \dots & 4X^2+4X+4 \end{array} \\ \begin{pmatrix} 1 & 0 & \dots & 1 & \dots & 0 \\ 0 & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & \dots & 1 \end{pmatrix} \end{array} = \mathcal{M}$$

Example 6 (BGLS)**.** For a concrete example of an underlying aggregate signature scheme $\Sigma' = (\mathsf{Gen}', \mathsf{Sig}', \mathsf{Agg}', \mathsf{Verify}')$, we consider the aggregate signature scheme of Boneh, Gentry, Lynn and Shacham (BGLS) [BGLS03].

BGLS describes a bilinear aggregate signature scheme based on CDH, which is EUF-CMA-secure in the random oracle model. The system parameters of BGLS are $(\mathbb{G}, g, G_T, p, e) \leftarrow \mathsf{Grp}(1^k)$ generated as in chapter 2, where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order p and e is an efficiently computable non-degenerate bilinear map. The scheme employs a full-domain hash function $H : \{0,1\}^* \to \mathbb{G}$ with message space $\mathcal{M}_k = \{0,1\}^*$ and requires that all messages in an aggregate signature are distinct.

Key Generation. $\mathsf{Gen}'(1^k)$, on input 1^k, chooses $x \leftarrow \mathbb{Z}_p$, and outputs a key pair $(pk, sk) := (g^x, x)$

Sign. $\mathsf{Sig}'(sk, M)$, on input a secret key $sk = x$ and a message $M \in \{0,1\}^*$, computes $\sigma := H(M)^x$, and outputs σ.

Aggregation. $\mathsf{Agg}'(C_1, C_2, \tau_1, \tau_2)$, on input exclusivley mergeable claim sequences $C_1 = (c_1, \ldots, c_r)$, $C_2 = (c_1, \ldots, c_s)$ (w.l.o.g $r < s$), (aggregate) signatures τ_1 and τ_2, computes $C := C_1 \sqcup C_2 = (c_1, \ldots, c_s)$ and $\tau := \tau_1 \cdot \tau_2 = \prod_{c_i \neq \perp}^{r} \sigma_i \cdot \prod_{c_i \neq \perp}^{s} \sigma_i$ and outputs (C, τ) (where $\sigma_i = H(M_i)^{x_i}$ for $c_i = (g^{x_i}, M_i) \neq \perp$).

(Aggregate) Verification. $\mathsf{Verify}'(C, \tau)$, on input a claim sequence $C = (c_1, \ldots, c_n)$ and an (aggregate) signature $\tau = \prod_{c_i \neq \perp}^{n} \sigma_i$, computes $h_i := H(M_i)$, for $M_i \in c_i \neq \perp$, and outputs 1 if $e(\tau, g) = \prod_{c_i \neq \perp}^{n} e(h_i, g^{x_i})$, else 0.

We construct from this a fault-tolerant aggregate signature scheme (with list verification) $\Sigma_{ft} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Agg}, \mathsf{Verify})$ as described in our construction in section 4.4 (especially of the algorithms Agg and Verify).

We only want to show here our statement of Remark 3, maintains that the multiset $\mathcal{R}$ of all claims corresponding to a regular signature is not necessarily equal to the multiset $\mathcal{C}_{\text{valid}}$ outputted by Verify (i.e. $\mathcal{R} \subseteq \mathcal{C}_{\text{valid}} = \mathsf{Verify}(C, \tau)$).

For this we set the parameters $d = 2, m = 25$ and $n = 125$ (as above in Example 5) to achieve a 2-CFF. For BGLS as the underlying aggregate signature scheme we obtain

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_{25} \end{pmatrix} = \begin{pmatrix} \sigma_1 \cdot \sigma_5 \cdot \ldots \\ \sigma_2 \cdot \sigma_6 \cdot \ldots \\ \vdots \\ \sigma_5 \cdot \ldots \cdot \sigma_{125} \end{pmatrix} \hat{=} \begin{pmatrix} \sigma_1 & 0 & \ldots & \sigma_5 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ldots & 0 & \sigma_6 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_5 & 0 & \ldots & \sigma_{125} \end{pmatrix},$$

where $\sigma_i \in \mathbb{G}$, $i = 1, \ldots, 125$, and $\tau \in \mathbb{G}^{25}$.
For $d = 2$ faulty signatures the scheme will still be able to output all correctly signed messages. Suppose for instance, that σ_1 is the valid signature for c_1 but we add a faulty signature $\tilde{\sigma_1}$ for c_1 in column 1 instead of σ_1, which we can write as

$$\tilde{\sigma_1} = g^r \cdot \sigma_1 \in \mathbb{G}, \text{ for some } r \in \mathbb{Z}_p.$$

If there is another faulty signature, e.g

$$\tilde{\sigma_5} = g^{r'} \cdot \sigma_5, \text{ for some } r' \in \mathbb{Z}_p,$$

this leads to

$$\begin{aligned} \tilde{\tau_1} &= g^r \cdot \sigma_1 \cdot g^{r'} \cdot \sigma_5 \cdot \ldots \\ &= g^{r+r'} \cdot \sigma_1 \cdot \sigma_5 \cdot \ldots \end{aligned}$$

in row 1. For $r' = -r$ we have

$$\begin{aligned} \tilde{\tau_1} &= g^{r-r} \cdot \sigma_1 \cdot \sigma_5 \cdot \ldots \\ &= \sigma_1 \cdot \sigma_5 \cdot \ldots \\ &= \tau_1 \end{aligned}$$

In this case, row 1 would also be valid, i.e $\mathsf{Verify}'(C[\mathcal{M}_1], \tilde{\tau_1}) = 1$. Although the signatures $\tilde{\sigma_1}$ and $\tilde{\sigma_5}$, respectiveley, are not regular for the corresponding claims c_1 and c_5, respectivley, these claims will also be added to $\mathcal{C}_{\text{valid}}$. In this case the set $\mathcal{R}$ of regular claims is a proper set of the ouput of Verify, i.e.

$$\mathcal{R} \subset \mathcal{C}_{\text{valid}}$$

Note, that this only happens with negligible probability.

Compression Ratio of Our Bounded Scheme. If our bounded scheme is instantiated with this polynomial-based CFF, and we assume that the length of signatures of the underlying scheme Σ' is bounded by a constant in n (but poly in k) *const*, then, as shown in Equation 4.1, the compression ratio is

$$\rho(n) = \frac{n}{\text{rows}(\mathcal{M})} = \frac{n}{|\mathcal{S}|} = \frac{n}{q^2} \ .$$

For $n = |\mathcal{B}|$, we therefore have

$$\rho(n) = \frac{|\mathcal{B}|}{|\mathcal{S}|} = \frac{q^{l+1}}{q^2} \ .$$

Since $q \geq dl + 1$, we have that $|\mathcal{B}|$ grows exponentially in l, whereas $|\mathcal{S}|$ grows only quadratically in l. Hence, $|\mathcal{B}|$ is exponential in $|\mathcal{S}|$, or, stated differently, $|\mathcal{S}|$ is logarithmic in $|\mathcal{B}|$.

Compression Ratio of Our Unbounded Scheme. When our unbounded scheme is instantiated with the monotone family of CFFs obtained by fixing an incidence matrix $\mathcal{M}$ and repeatedly using Lemma 11 on $\mathcal{M}$, then the asymptotic compression ratio is $\rho(n) = 1$, since

$$\frac{n}{\text{rows}(\lambda)} \leq \frac{\text{cols}(\lambda)}{\text{rows}(\lambda)} = \frac{\text{cols}(\mathcal{M})}{\text{rows}(\mathcal{M})} \quad \text{for all } \lambda,$$

which is constant. Therefore, the size of an aggregate signature is linear in the length of the claim sequence.

However, if we assume that all signatures of the underlying scheme Σ' have a size bounded by *const*, then the concrete size of an aggregate signature is at most

$$\begin{aligned} \text{rows}(\lambda) \cdot \mathit{const} &\leq \lambda \cdot \text{rows}(\mathcal{M}) \cdot V \\ &\leq (n/\text{cols}(\mathcal{M}) + 1)\,\text{rows}(\mathcal{M}) \cdot \mathit{const} \\ &= \left(\frac{\text{rows}(\mathcal{M})}{\text{cols}(\mathcal{M})} n + \text{rows}(\mathcal{M})\right) \cdot \mathit{const} \ , \end{aligned}$$

since $\text{rows}(\lambda) = \lambda \cdot \text{rows}(\mathcal{M})$ for the construction of the monotone family of CFFs, and $\lambda = \lceil n/\text{cols}(\mathcal{M}) \rceil \leq n/\text{cols}(\mathcal{M}) + 1$.

Therefore the length of the aggregate signature is linear in n, but the factor $\text{rows}(\mathcal{M})/\text{cols}(\mathcal{M})$ can be made arbitrarily small by choosing a proper CFF, such as the one described above.

It is an interesting open problem to construct an unbounded fault-tolerant scheme with better compression ratio, for example by finding a better monotone family of CFFs. A generalization of the above construction to multivariate polynomials, which might be advantageous in some scenarios, is given in section 4.5.3.

4.5.2 Selective Verification

With this univariate polynomial-based construction of a d-CFF it is very easy to generate our incidence matrix or only some parts of it, which we need for our verification algorithm or if we want to check some information separately.

If, for example, one is interested to verify the validity of only one single claim signature pair (c_j, σ_j) in an aggregate signature, it is not necessary to generate the whole matrix but only the rows where the related column j has 1-entries. So, you only have to know which polynomial corresponds to column j.
For this, we can use the fact, that for each positive number

$$n \in \{0, \ldots, q^{l+1} - 1\}$$

exists a unique q-adic representation, i.e.

$$n = a_l \cdot q^l + a_{l-1} \cdot q^{l-1} + \cdots + a_0, \text{ where } a_l, \ldots, a_0 \in \{0, \ldots, q-1\}.$$

So, each n corresponds to a $(l+1)$-tuple denoted by

$$(a_l^{(n)}, \ldots, a_0^{(n)}).$$

Thus, for column $j \in \{0, \ldots, q^{l+1}-1\}$ we assign the polynomial

$$f_j = a_l^{(j)} X^l + \cdots + a_0^{(j)}.$$

Analogously, for each row $i = 0, \ldots, q^2 - 1$,we assign the tuple

$$(b_1^{(i)}, b_0^{(i)}) \in \mathbb{F}_q^2, \text{ where } i = b_1^{(i)} \cdot q + b_0^{(i)}.$$

Let

$$I'_j := \left\{ i \in \{0, \ldots q^2 - 1\} : f_j(b_1^{(i)}) = b_0^{(i)} \right\}, \text{ for } j \in \{0, \ldots, q^{l+1} - 1\},$$

be the subsets of all rows $i' \in I'_j$ where $f_j(b_1^{(i')}) = b_0^{(i')}$ for corresponding $j = 0, \ldots, q^{l+1} - 1$.

Hence, it suffices to generate only the rows $i' \in I'_j$ to verify the validity of σ_j. To get the 1-entries of these rows, we have to check for each $i' \in I'_j$ which polynomials $f \in \mathbb{F}_q[X]_{\leq l}$ satisfy $f(b_1^{(i')}) = b_0^{(i')}$. For all arbitrary, but fixed values $a_l, \ldots, a_1 \in \{0, \ldots, q-1\}$ compute an appropriate a_0. This results in q^l polynomials, accordingly columns, per row. If the coefficients of the appropriate polynomials are known then we can use them to compute the number of the corresponding columns with 1-entries, as described before.

4.5.3 Cover-Free Family Based on Multivariate Polynomials

For our polynomial based construction, we can also use multivariate polynomials

$$f \in \mathbb{F}_q[X_1, \ldots, X_t], \ t \in \mathbb{N},$$

of degree at most l. Each multivariate polynomial f with degree $\leq l$ consists of monomials in terms of

$$a_{i_1,\ldots,i_t} X_1^{i_1} \cdots X_t^{i_t}, \text{ where } a_{i_1,\ldots,i_t} \in \mathbb{F}_q \text{ and } i_1 + \cdots + i_t \leq l.$$

We denote by

$$\mathbb{F}_q[X_1, \ldots, X_t]_{\leq l} := \left\{ \sum_{i_1+\cdots+i_n \leq l} a_{i_1,\ldots,i_t} X_1^{i_1} \cdots X_t^{i_t} \colon a_{i_1,\ldots,i_t} \in \mathbb{F}_q \right\}$$

the set of all multivariate polynomials $f \in \mathbb{F}_q[X_1, \ldots, X_t]$ of degree at most l. The maximal number of monomials of degree exactly l, is

$$\binom{t+l-1}{l}.$$

Hence, for degree at most l, we have

$$\sum_{i=0}^{l} \binom{t+i-1}{i} = \binom{t+l}{l} \text{ and hence, } |\mathbb{F}_q[X_1, \ldots, X_t]_{\leq l}| = q^{\binom{t+l}{l}}.$$

We can now define

$$B_f := \left\{ (\mathbf{x}, \mathbf{f}(\mathbf{x})) \colon \mathbf{x} \in \mathbb{F}_{\mathbf{q}}^{\mathbf{t}} \right\} \text{ with } |B_f| = q^t,$$

and

$$\mathcal{B} := \{ B_f \colon f \in \mathbb{F}_q[X_1, \ldots, X_t]_{\leq l} \} \text{ with } |\mathcal{B}| = q^{\binom{t+l}{l}}.$$

Now, we set

$$\mathcal{S} := \mathbb{F}_q^{t+1}, \text{ which is of size } |\mathcal{S}| = q^{t+1}.$$

The number of zeros is at most $l \cdot q^{t-1}$ and thus, for different $B_f, B_{f_1}, \ldots, B_{f_d} \in \mathcal{B}$ it holds

$$\left| B_f \setminus \bigcup_{i=1}^{d} B_{f_i} \right| \geq q^t - d \cdot l \cdot q^{t-1}.$$

To achieve a d-CFF with these definitions, the condition,

$$q^t \geq d \cdot l \cdot q^{t-1} + 1$$

must be satisfied.

Compression Ratio of Our Bounded Scheme. If our bounded scheme is instantiated with this multivariate CFF, and we assume for simplicity, that the size of signatures of the underlying scheme Σ' is bounded by a constant, then as shown in Equation 4.1 , the compression ratio is

$$\rho(n) = \frac{n}{\text{rows}(\mathcal{M})} = \frac{n}{|\mathcal{S}|} = \frac{n}{q^{t+1}} \ .$$

For $n = |\mathcal{B}|$, we therefore have

$$\rho(n) = \frac{|\mathcal{B}|}{|\mathcal{S}|} = \frac{q^{\binom{t+l}{l}}}{q^{t+1}} \ .$$

Compression Ratio of Our Unbounded Scheme. By using Lemma 11 on $\mathcal{M}$ we can also obtain a monotone CFF based on multivariate polynomials and use it to instantiate our unbounded scheme. The discussion about the compression ratio of the unbounded scheme in the previous section also applies to this instantiation.

4.6 State-of-the-Art

First, we want to show an application of our d-fault-tolerant aggregate signature scheme, which was a follow-up paper published in [HKKKH17]. We emphasize here, that this is not part of this thesis and for more details we refer to [HKKKH17].

4.6.1 Application of Fault-Tolerant Sequential Aggregate Signatures

Aggregate signature schemes have another important application in the field of secure logging. Log files are used to record events like user actions, system errors, failed log-in attempts as well as general information, and play an important role in computer security by providing, for example, accountability and a basis for intrusion detection. Log files are usually kept for very long periods of time, which means that thousands or even millions of log entries need to be stored. Keeping correct and informative log files is crucial for system maintenance, security and forensics. Cryptographic logging schemes offer integrity checks that protect a log file even in the case where an adversary has broken into the system.

A relatively recent feature of these schemes is resistance against *truncations*, i.e. the deletion and/or replacement of the end of the log file. This is especially relevant as system intruders are typically interested in manipulating the later log entries that point towards their attack. However, there are not many schemes that are resistant against truncating the log file, and those that are, have at least one of the following disadvantages: They are *memory intensive* (they store at least one signature per log entry), or *fragile*, i.e. a single error in the log renders the signature invalid and useless in determining where the error occurred.

In [HKKKH17] we constructed a publicly-verifiable secure logging scheme that is robust (not fragile), space efficient and truncation secure under simple assumptions. Our generic construction uses a variant of fault-tolerant aggregate signature scheme, which is sequential and forward-secure (a specific security notion). Because of the fault-tolerance it can cope with a number of manipulated log entries (bounded a priori) and offer strong robustness guarantees while still being space efficient.

4.6.2 Recent Improvements

In [HKKKR16] we pose the open problem to find a better monotone family, for the case of unbounded aggregation, in order to achieve a more efficient unbounded scheme. In [BIM18] they define a *nested* cover-free family, which is a more flexible sequence of d-cover-free families. They provide concrete constructions of such families that yield unbounded fault-tolerant aggregate signature schemes with a better compression ratio than in [HKKKR16].

Another approach for constructing fault-tolerant aggregate signature schemes instead of applying a cover-free family is given in [WCD18]. They make use of concepts from finite set theory and put forward a new method to separate all individual signatures in many subsets. Then, verifiying the valid individual signatures becomes more efficient. Compared to [HKKKR16] their scheme can verify more valid individual signatures while verifying less aggregated signatures.

5 Almost Tight Identity-Based-Encryption Security

In this chapter we focus on identity-based encryption (IBE) schemes. Informally, an IBE scheme is a variant of a public-key encryption scheme in which the public key of a user is some information about his identity (e.g. an e-mail address). Together with a master public key and some public parameters, this allows encrypting without distributing all user public keys in advance. A user obtains a (user) secret key from a trusted authority, which only needs the identity and a master secret key. A generic transformation from any IBE scheme to a digital signature scheme exist [BF01].

We instantiate an (almost) tightly and fully secure identity-based encryption (IBE) scheme in the multi-instance, multi-ciphertext setting under a simple assumption. We recall in more detail in the following what we mean with tight security, especially in the case of a multi-instance, multi-ciphertext scenario.

We emphasize here, that we almost entirely taken the important parts of these chapter from [HKS15], partly verbatim, with some useful additions, descriptions and further explanations.

Tight Security in General and the Multi-Instance, Multi-Ciphertext Setting.

For many cryptographic primitives (e.g encryption or signature schemes), we currently cannot prove security directly. Hence, we typically *reduce* the security of a given scheme to the hardness of a computational problem, in the sense that every successful and efficient adversary A on the scheme, can be used to construct a successful and efficient problem solver B. Thus, B solves a computational problem with significant probability, which is assumed to be not efficiently solvable. This contradiction shows that there cannot exist such an efficient and successful adversary A on the scheme, which proves the security of the scheme.

Usually, B has a lower probability of success than A. Frequently, the reason for this is that B has to guess something, e.g. where he embeds the computational problem. Now it may be that the probability of success of B decreases inversely proportional to the number of ciphertexts or signatures. To look at this *loss* of such a reduction is both a theoretically and practically interesting question.

Informally, the loss of a reduction quantifies the difference between the success of a hypothetical adversary A on the cryptographic scheme, and the success of the derived problem solver B. The smaller the loss is, the tighter the security of the cryptographic scheme is related to the underlying computational problem. Therefore, it is desirable to develop primitives whose security can be reduced *tightly* (i.e., loss-freely) to the security of the underlying assumption, i.e., in which the success probability of B is independent of the number of encrypted or signed messages.

A tight reduction is also desirable from a practical perspective. Otherwise, if the cryptographic primitive is instantiated in concrete terms, it is necessary - in order to achieve sufficient security - that the parameters (e.g the key length) are set accordingly large, which has a negative impact on efficiency. In other words, the tighter a reduction, the better are the security guarantees we can give for a specific instance of the scheme.

Multi-Instance, Multi-Ciphertext Setting. However, in most practical usage scenarios, a cryptographic primitive is used multiple times. The multi-instance, multi-ciphertext setting represents a more realistic scenario. In this setting many instances of a scheme are used, which

represents many different users of the scheme or, e.g., in the IBE setting many different executions of the scheme for different master public keys and corresponding master secret keys. An adversary on the scheme is allowed to see many different challenge ciphertexts per instance. In the case of digital signatures, only the multi-instance setting (denoted as multi-user setting) is considered. Depending on the type of the scheme and the security notion corruptions are allowed or not, i.e., the adversary is allowed to learn different (user) secret keys or not. Hence, tight security reductions become particularly meaningful when they reduce an adversary on the whole system (with many instances of the cryptographic scheme) to a problem solver. For many primitives (such as secret-key [BDJR97] or public-key [BDPR98] encryption), one-instance security is known to imply multi-instance and multi-ciphertext security. However, [BBM00] shows that in general the security loss depends on the number of instances and number of challenge ciphertexts. Thus, this generic reduction is not tight and the corresponding security guarantees for concrete schemes may indeed vanish in the number of instances and challenge ciphertexts.

Existing Tightly Secure Schemes. The loss of security reductions has been considered explicitly by [BDJR97] for the case of encryption schemes. The first "somewhat tight" reductions (whose loss is independent of the number of instances of the scheme, but not of the number of ciphertexts) for public-key encryption (PKE) schemes were given in [BBM00]. In the following years, more tight (or somewhat tight) reductions for encryption schemes were constructed in the random oracle model [GMMV03; CKS08; Bol09], or from "q-type" assumptions [Gen06; GH09].[1]

However, only recently, the first PKE schemes emerged [HJ12; ADKNO13; LJYP14] whose tight security (in the multi-instance, multi-ciphertext setting) can be proved under simple assumptions in the standard model. Even more recently, identity-based encryption (IBE) schemes with "almost tight" security (under simple assumptions) have been constructed [CW13; BKP14]. This required new techniques, since it is not clear how to extend the techniques of [HJ12; ADKNO13; LJYP14] to the IBE setting. In this case, "almost tight" means that their security reduction loss only depends on the security parameter, but still considers the standard IBE security experiment [BF01] with one encryption and one instance of the scheme. Nonetheless, while the IBE schemes from [CW13; BKP14] are not proved tightly secure in a multi-user, multi-ciphertext setting, these schemes imply tightly secure PKE schemes (even in the multi-user, multi-ciphertext setting) when plugged into the transformations of [BF01; HJ12; LJYP14].

Our Contribution

We instantiate the first almost tightly and fully secure IBE scheme under a simple (dual system) assumption in composite-order pairing-friendly groups. This is also interesting regarding multi-instance (multi-user) fully secure signature schemes with an almost tight security reduction, since any IBE scheme can be converted into a signature scheme by the Naor transformation.

5.1 Organization

In section 5.3 we give a formal definition of an IBE scheme and consider security notions for the standard setting and for the multi-instance, multi-ciphertext setting.

We describe a known transformation from any IBE scheme to a signature scheme and discuss the security relations in section 5.3.1.

In section 5.3.2 we explain the dual system strategy for proving full security of IBE schemes introduced by Waters [Wat09] and some resulting variants. We mention nested dual system groups [CW13], which enable to prove the first almost tightly and fully secure IBE scheme in

[1] A "q-type" assumption may depend on the size of the investigated cryptographic system. (That is, larger cryptographic systems may only be secure under a stronger instance of the assumption.) Hence, a tight reduction (even in a multi-instance scenario) to a q-type assumption may not yield security guarantees that are independent of the number of users.

the one instance setting. We explain their high-level proof strategy and why it is not applicable for multi-instance and multi-ciphertext IBE schemes. This leads to extended nested dual system groups which are formally described in section 5.4 and necessary for proving the security in the multi-instance, multi-ciphertext setting. In section 5.4.1 the generic construction is given for completeness. We mention here that this is not part of this thesis and a further discussion on this topic can be found in [Str15].

In section 5.5 we give an instantiation of extended dual system groups which leads to an almost tightly and fully secure IBE scheme in the multi-instance, multi-ciphertext setting under a simple (dual system) assumption.

5.2 Preliminaries

For an algorithm $y \leftarrow A(1^k, x)$, on input a security parameter 1^k and x, and output y, we write

$$(y_i)_{i\in[n]} \leftarrow (A(1^k, x))^n, n \in \mathbb{N}$$

for executing $A(1^k, x)$ n times and obtaing n outputs $(y_1, \ldots, y_n)$.

We write vectors in bold font, e.g., $\mathbf{v} = (v_1, \ldots, v_n)$ for a vector of length $n \in \mathbb{N}$ and with components $v_1, \ldots, v_n$. (We may also write $\mathbf{v} = (v_i)_{i\in[n]}$ or even $\mathbf{v} = (v_i)_i$ if the dimension is clear from the context.) In the following, we use a *component-wise* multiplication of vectors, i.e.,

$$\mathbf{v} \cdot \mathbf{v}' = (v_1, \ldots, v_n) \cdot (v'_1, \ldots, v'_n) = (v_1 \cdot v'_1, \ldots, v_n \cdot v'_n).$$

Further, we write

$$\mathbf{v}^j := (v_1^j, \ldots, v_n^j),$$

for component-wise exponentiation with $j \in \mathbb{N}$.
To remove the entry at position i from a vector $\mathbf{v} = (v_i)_{i\in[n]}$ we write

$$\mathbf{v}_{-i} := (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n), \text{ for } i \in [n].$$

Finally, exponentiation with a vector $\mathbf{v} = (v_i)_{i\in[n]}$ is written as

$$s^{\mathbf{v}} := (s^{v_1}, \ldots, s^{v_n}).$$

The Bilinear Decisional Diffie-Hellman (BDDH) Problem. In section 5.4.1 and section 5.5 we introduce two variants of the Bilinear Decisional Diffie-Hellman (BDDH) assumption, which is defined in the following.

Definition 34 (BDDH assumption)**.** We say that the *Bilinear Decisional Diffie-Hellman (BDDH) assumption* holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$ if

$$\mathsf{Adv}^{\mathrm{bddh}}_{\mathsf{Grp},A}(k) := |\Pr\left[A(1^k, g, g^a, g^b, g^c, g^{abc}) = 1\right] - \Pr\left[A(1^k, g, g^a, g^b, g^c, g^d) = 1\right]|$$

is negligible for any PPT adversary A, where $(\mathbb{G}, g, p) \leftarrow \mathsf{Grp}(1^k)$ and $a, b, c, d \leftarrow \mathbb{Z}_p$ are uniformly chosen.

5.3 Identity-Based Encryption (IBE)

In the following we give a formal definition of an identity-based encryption (IBE) scheme and security notions in the one-instance, one-ciphertext and multi-instance, multi-ciphertext scenario, respectively. Furthermore, we describe the relation to signature schemes and some important techniques for proving security and its difficulties.

Definition 35 (Identity-based encryption)**.** An *identity-based encryption* (IBE) scheme $\mathsf{IBE} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ with identity space $\mathcal{ID}$ and message space $\mathcal{M}_k$ consists of five PPT algorithms.

Parameter Sampling. $\mathsf{Par}(1^k, n)$, on input a security parameter 1^k and an identity length parameter $n \in \mathbb{N}$, outputs public parameters pp and secret parameters sp. (We assume that Ext, Enc, and Dec have implicitly access to pp.)

Key Generation. $\mathsf{Gen}(pp, sp)$, on input public parameters pp and secret parameters sp, outputs a master public key mpk and a master secret key msk.

User Secret Key Extraction. $\mathsf{Ext}(msk, id)$, on input a master secret key msk and an identity $id \in \mathcal{ID}$, outputs a user secret key usk_{id} associated with id.

Encryption $\mathsf{Enc}(mpk, id, M)$, on input a master public key mpk, an identity $id \in \mathcal{ID}$, and a message $M \in \mathcal{M}_k$, outputs a ciphertext C_{id} associated with id.

Decryption $\mathsf{Dec}(usk_{id}, C_{id})$, on input a user secret key usk_{id} for an identity $id \in \mathcal{ID}$, and a ciphertext C_{id}, outputs $M \in \mathcal{M}_k \cup \{\perp\}$.

We require IBE to be *correct* in the sense that for any $k, n \in \mathbb{N}$, for all $(pp, sp) \leftarrow \mathsf{Par}(1^k, n)$, for all $(mpk, msk) \leftarrow \mathsf{Gen}(pp, sp)$, for all $id \in \mathcal{ID}$, for all $usk_{id} \leftarrow \mathsf{Ext}(msk, id)$, for all $M \in \mathcal{M}_k$, and for all $C_{id} \leftarrow \mathsf{Enc}(mpk, id, M)$, Dec satisfies $\mathsf{Dec}(usk_{id}, C_{id}) = M$.

Experiment $\mathsf{Exp}^{\text{ibe-ind-cpa}}_{\mathsf{IBE},A}(k, n)$
$(pp, sp) \leftarrow \mathsf{Par}(1^k, n)$
$(mpk, msk) \leftarrow \mathsf{Gen}(pp, sp)$
$(id^*, M_0^*, M_1^*) \leftarrow A^{\mathsf{Ext}(msk,\cdot)}(pp, mpk)$
$b \leftarrow \{0, 1\}$
$C^*_{id^*} \leftarrow \mathsf{Enc}(mpk, id^*, M_b^*)$
$b^* \leftarrow A^{\mathsf{Ext}(msk,\cdot)}(C^*_{id^*})$
if $b = b^*$ **and** $|M_0^*| = |M_1^*|$
and A has not queried $\mathsf{Ext}(msk, id^*)$
return 1
else
return 0

Figure 5.1: The IBE-IND-CPA experiment.

Security Notion for IBE. The security notion for an IBE scheme [BF01], dubbed IBE-IND-CPA security, and the experiment is defined in Definition 36, and briefly described here:

- The experiment generates some parameter pair $(pp, sp) \leftarrow \mathsf{Par}(1^k, n)$, a master public and secret key pair $(mpk, msk) \leftarrow \mathsf{Gen}(pp, sp)$, and provides A with (pp, mpk).
- During the experiment, A has access to an $\mathsf{Ext}(msk, \cdot)$-oracle to adaptively query user secret keys for identities $id \in \mathcal{ID}$.
- Subsequently, A outputs a challenge identity $id^* \in \mathcal{ID}$ and two messages $M_0^*, M_1^* \in \mathcal{M}$.
- The experiment then computes $C^*_{id^*} \leftarrow \mathsf{Enc}(mpk, id^*, M_b^*)$, for $b \leftarrow \{0, 1\}$, and sends the ciphertext $C^*_{id^*}$ to A.
- Finally, A outputs a guess b^*.

The adversary A *wins* the experiment if $b = b^*$, $|M_0^*| = |M_1^*|$ and id^* has never been submitted to the Ext-oracle.

Definition 36 (IBE-IND-CPA)**.** An IBE scheme IBE is *indistinguishable under chosen plaintext attacks* (short: IBE-IND-CPA-secure) if

$$\mathsf{Adv}^{\text{ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n) := |\Pr\left[\mathsf{Exp}^{\text{ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n) = 1\right] - 1/2\,|$$

is negligible for any PPT adversary A. Experiment $\mathsf{Exp}^{\text{ibe-ind-cpa}}_{\mathsf{IBE},A}$ is defined in Figure 5.1.

Multi-Instance, Multi-Ciphertext Setting

The classical security model for IBE schemes described above requires that the *single* challenge ciphertext in a *single* instance of the scheme reveals nothing about the corresponding message even if the adversary can query user secret keys for up to q identities.

Now, we consider the scenario, where we have many instances of an IBE scheme, i.e. many master public keys, and in each instance the adversary obtains many challenge ciphertexts to distinguish for different identities, but for the same chosen bit b. This models a much more realistic scenario, which is very desirable.

Security Notion for Multi-Instance, Multi-Ciphertext IBE. The security notion for a multi-instance, multi-ciphertext IBE scheme, dubbed (μ, q)-IBE-IND-CPA security, for μ instances, and q challenge ciphertexts, where $(\mu, q) \in \mathbb{N}^2$, and the experiment is defined in Definition 37, and briefly described here:

Let $\mathsf{Enc}'(mpk, id, b, M_0, M_1)$ be a PPT auxiliary encryption oracle that, given a master public key mpk, a challenge identity $id \in \mathcal{ID}$, a bit $b \in \{0,1\}$, and two messages $M_0, M_1 \in \mathcal{M}$, outputs a challenge ciphertext $C_{id} \leftarrow \mathsf{Enc}(mpk, id, M_b)$.

- The experiment generates some parameter pair $(pp, sp) \leftarrow \mathsf{Par}(1^k, n)$, master public and secret key pairs $(mpk_j, msk_j) \leftarrow (\mathsf{Gen}(pp, sp))^\mu$, and provides A with (pp, mpk_j), for $j \in [\mu]$.
- During the experiment, A has access to an $\mathsf{Ext}(msk_j, \cdot)$-oracle, to adaptively query user secret keys for identities $id \in \mathcal{ID}$ under mpk_j, for all instances $j \in [\mu]$.
- A has also access to an $\mathsf{Enc}'(mpk_j, \cdot, b, \cdot, \cdot)$-oracle, to adaptively query challenge ciphertexts for corresponding mpk_j and a (uniform) bit $b \leftarrow \{0,1\}$, for all instances $j \in [\mu]$.
- Finally, A outputs a guess b^*.

The adversary A *wins* the experiment iff $b = b^*$ and A is *admissible*, i.e.

- A never queries an $\mathsf{Ext}(msk_j, \cdot)$ oracle on an identity id for which it has already queried the corresponding $\mathsf{Enc}'(mpk_j, \cdot, b, \cdot, \cdot)$ oracle (and vice versa)
- Each message pair A selected as input to Enc' contained only equal-length messages
- A has only queried its Enc'-oracles at most q times per j-instance.

Definition 37 ((Weak) (μ, q)-IBE-IND-CPA)**.** A multi-instance, multi-ciphertext IBE scheme IBE is *(μ,q)-indistinguishable under chosen plaintext attacks* (short: (μ, q)-IBE-IND-CPA-secure) if

$$\mathsf{Adv}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n) := |\Pr\left[\mathsf{Exp}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n) = 1\right] - 1/2\,|.$$

is negligible for any PPT adversary A. Experiment $\mathsf{Exp}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}$ is defined in Figure 5.2.

Furthermore, we call IBE *weak* (μ,q)*-indistinguishable under chosen plaintext attacks* (short: weak(μ, q)-IBE-IND-CPA-secure) if

$$\mathsf{Adv}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n) := |\Pr\left[\mathsf{Exp}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n)=1\right] - 1/2\,|.$$

is negligible for any *weak* PPT adversaries A. We call A *weak* if it never requests challenge ciphertexts for the same scheme instance and identity twice (i.e., if it never queries any $\mathsf{Enc}'(mpk_j,\cdot,b,\cdot,\cdot)$ oracle twice with the same identity id).

Experiment $\mathsf{Exp}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n)$
$(pp, sp) \leftarrow \mathsf{Par}(1^k, n)$
$(mpk_j, msk_j)_{j\in[\mu]} \leftarrow (\mathsf{Gen}(pp, sp))^\mu$
$b \leftarrow \{0,1\}$
$b^* \leftarrow A^{(\mathsf{Ext}(msk_j,\cdot),\mathsf{Enc}'(mpk_j,\cdot,b,\cdot,\cdot))_{j\in[\mu]}}(pp,(mpk_j)_{j\in[\mu]})$
if A is *admissible* **and** $b = b^*$
  return 1
else
  return 0

Figure 5.2: The (μ, q)-IBE-IND-CPA security experiment.

Remark 5. We remark that the one-instance, one-ciphertext notion $(1, 1)$-IBE-IND-CPA is the standard notion of IBE security as defined in Definition 36.

Note, that in general, an IBE scheme which is secure in the classical *single* model implies to be secure in the multi-instance, multi-ciphertext model. However, this implication is not tightness-preserving. If μ and q are the number of instances and challenge ciphertexts per instance, respectively, the generic reduction results in a multiplicative security loss $\mathbf{O}(\mu \cdot q)$ [BBM00].

5.3.1 Naor Transformation: From IBE to Signatures

Boneh and Franklin mentioned in [BF01] an observation by Naor: *"Any IBE scheme can be immediately converted into a public key signature scheme."*

The Naor transformation applied to their IBE scheme resulted in the BLS signature scheme with short signatures [BLS04]. Furthermore, Waters [Wat05] presented the first efficient IBE scheme that is fully secure without random oracles (a modification of the Boneh-Boyen scheme [BB04a]) and a resulting signature scheme via the Naor transformation that is EUF-CMA-secure under the CDH assumption without random oracles.

Further, the signature schemes thus derived from [CW13; BKP14] are then suitable for the conversions of [HJ12; LJYP14], yielding PKE schemes tightly secure in the multi-user, multi-ciphertext setting.

Naor Transformation. Let $\mathsf{IBE} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ be an identity-based encryption scheme with identity space $\mathcal{ID} = \{0,1\}^n$ and message space $\mathcal{M}_k$. For $(pp, sp) \leftarrow \mathsf{Par}(1^k, n)$, the signature scheme $\Sigma_{\mathsf{NT}} = (\mathsf{Gen}_{\mathsf{NT}}, \mathsf{Sig}_{\mathsf{NT}}, \mathsf{Ver}_{\mathsf{NT}})$, with message space $M^\Sigma_k = \mathcal{ID}$, consists of the following three algorithms (see also Figure 5.3):

Key Generation. $\mathsf{Gen}_{\mathsf{NT}}(pp, sp)$, on input public parameters pp and secret parameters sp, runs $(mpk, msk) \leftarrow \mathsf{Gen}(pp, sp)$, and outputs key pair $(pk, sk) := (mpk, msk)$.

Signing. $\mathsf{Sig}_{\mathsf{NT}}(sk, M)$, on input a secret key sk and a message $M \in \mathcal{M}^\Sigma_k = \mathcal{ID}$, computes $usk_M \leftarrow \mathsf{Ext}(sk, M)$ and outputs a signature $\sigma = usk_M$.

$\mathsf{Gen}_{\mathsf{NT}}(pp, sp)$	$\mathsf{Sig}_{\mathsf{NT}}(sk, M)$	$\mathsf{Ver}_{\mathsf{NT}}(pk, M, \sigma = usk_M)$
$(mpk, msk) \leftarrow \mathsf{Gen}(pp, sp)$ $pk := mpk$ $sk := msk$ **return** (pk, sk)	$usk_M \leftarrow \mathsf{Ext}(sk, M)$ **return** $\sigma := usk_M$	choose random $M' \in \mathcal{M}_k$ $C_M \leftarrow \mathsf{Enc}(pk, M, M')$ $\tilde{M} = \mathsf{Dec}(\sigma, C_M)$ **if** $M' = \tilde{M}$ **return** 1 **else** **return** 0

Figure 5.3: Naor transformation applied to an identity-based encryption scheme $\mathsf{IBE} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ to obtain the signature scheme $\Sigma_{\mathsf{NT}} = (\mathsf{Gen}_{\mathsf{NT}}, \mathsf{Sig}_{\mathsf{NT}}, \mathsf{Ver}_{\mathsf{NT}})$, with message space $\mathcal{M}_k^{\Sigma} = \mathcal{ID} = \{0,1\}^n$.

Verification. $\mathsf{Ver}_{\mathsf{NT}}(pk, M, \sigma)$, on input a public key pk, a message M and a purported signature σ, chooses $M' \leftarrow \mathcal{M}_k$ randomly from the message space of IBE, computes $C_M \leftarrow \mathsf{Enc}(pk, M, M')$, decrypts $\tilde{M} = \mathsf{Dec}(\sigma, C_M)$, and ouputs 1 if $M' = \tilde{M}$, else 0.

Note, that the verification algorithm $\mathsf{Ver}_{\mathsf{NT}}$ of the signature scheme Σ_{NT} is randomized.

Relations between Security Notions of IBE Schemes and Signature Schemes. In [CFHIZ07] they give formal security treatments for the Naor transformation and discuss several implications and separations among security notions of identity-based encryption schemes and transformed signature schemes.

Their main result states that an IBE-IND-CPA-secure identity-based encryption scheme with messages space $\mathcal{M}_k$, such that $\frac{1}{|\mathcal{M}_k|}$ is negligible in k (e.g. $\mathcal{M}_k = \{0,1\}^k$), already implies an EUF-CMA-secure transformed signature scheme. Therefore, secure signature schemes can be generally derived from IBE schemes.

Multi-User Signature Schemes.

In this chapter, we construct an IBE scheme in the multi- instance, multi-ciphertext setting with security loss $\mathbf{O}(k)$ independent of the number of instances, challenge ciphertexts and user secret key queries. Therefore, it is important to consider the relation to multi-user signature schemes, which we obtain after applying the Naor transformation.

The multi-user setting describes a more realistic scenario for many applications of digital signature schemes in the real world [HJ12; Bad14]. Informally, in a multi-user signature scheme the security experiment allows an adversary to query signatures from different users, i.e. signed under different secret keys for which he knows the corresponding public keys. Then, he has to produce a forgery (M^*, σ^*), which is valid under a public-key pk_j for one of the users and for a fresh message M^*, which he never queried corresponding to sk_j, $j \in [\mu]$, where μ is the number of instances, i.e. users in the system. The following definition is the natural extension of EUF-CMA security in the multi-user setting.

Definition 38 (Multi-user existential unforgeability under chosen-message attacks)**.** We say a signature scheme is *existentially unforgeable under chosen-message attacks in the multi-user setting* (μ-EUF-CMA-secure), where μ is the number of users, if

$$\mathsf{Adv}_{\Sigma,F}^{\mu\text{-euf-cma}}(k) := \Pr\left[\mathsf{Exp}_{\Sigma,F}^{\mu\text{-euf-cma}}(k) = 1\right]$$

is negligible for any PPT adversary F, where $\mathsf{Exp}_{\Sigma,F}^{\mu\text{-euf-cma}}(k)$ is defined in Figure 5.4.

Experiment $\mathsf{Exp}^{\mu\text{-euf-cma}}_{\Sigma,F}(k)$
$(pk_j, sk_j)_{j\in[\mu]} \leftarrow (\mathsf{Gen}(pp, sp))^{\mu}$
$(M^*, \sigma^*) \leftarrow F^{\mathsf{Sig}(\cdot,\cdot)}(pk_1, \ldots, pk_\mu)$
if $\exists j^* \in [\mu] : \mathsf{Ver}(pk_{j^*}, M^*, \sigma^*) = 1$
  and F has not queried $\mathsf{Sig}(sk_{j^*}, M^*)$
  return 1
else
  return 0

Figure 5.4: The μ-EUF-CMA security experiment, where $\mu \in \mathbb{N}$ is the number of users.

Problems in Constructing Tightly Secure (Multi-User) Signature Schemes. It is a straightforward reduction to show that EUF-CMA security implies μ-EUF-CMA security, by just simply guessing the appropriate public key and corresponding secret key under which the adversary will forge a signature. Therefore the security loss is linear in the number of users, i.e. $\mathbf{O}(\mu)$ (similar to IBE schemes in the multi-instance setting). A lot of recent work focused on constructing tightly secure signature schemes [KW03; Sch11; HJ12; HJK12; Seu12; FJS14]. However, there were only few in the multi-user setting [HJ12; Bad14; BHJKL15] at that time. In [HJ12] they construct a tightly μ-EUF-CMA-secure structure-preserving signature scheme in the standard model under a standard assumption. Informally, in a structure-preserving signature scheme all operations can be expressed using equations over a (usually pairing-friendly) cyclic group. However, the signatures of their scheme are very large. This inefficiency of the parameters is often a result of tight security reductions, especially in the context of structure-preserving signatures.

Hence, due to the relation between IBE schemes and signature schemes our result of an almost tightly secure IBE scheme in the multi-instance, multi-ciphertext setting is also interesting regarding the construction of tightly secure multi-user signature schemes.

In 2018, [GJ18] considered an extension of μ-EUF-CMA security, dubbed μ-EUF-CMA^{corr}, where an adversary is allowed to corrupt users and thus obtains the corresponding secret keys. They constructed the first practical tightly μ-EUF-CMA^{corr}-secure signature scheme in the standard model without bilinear maps.

5.3.2 The Development of IBE and Dual System Methodology

The Dual System Methodology established a new way to prove security of IBE schemes and related cryptographic schemes. This proof strategy was developed and published by Waters [Wat09]. Before we explain this strategy on a high-level we briefly summarize some previous ideas, which led to the new approach.

The first construction of an efficient IBE scheme was introduced by Boneh and Franklin [BF01] using bilinear maps. Their proof of security is in the random oracle model under the Bilinear Diffie-Hellman assumption. One important question was, if it is possible to prove security of their scheme in the standard model. Canetti, Halevi and Katz [CHK07] proved security of the Boneh-Franklin IBE scheme without a random oracle, but in a weaker model, called the *selective-ID model*. In this model an adversary has to choose his challenge identity id^* before he even sees the master public key and public parameters of the system. Boneh and Boyen [BB04b] constructed the first fully secure IBE scheme in the standard model, and subsequently Waters [Wat05] published one with better efficiency. All of these works used a similar proof strategy, namely partitioning.

Partitioning Strategy. In a partitioning reduction, as in general reductions, one reduces the security of the scheme to an underlying complexity assumption. Here, partitioning means, that a reduction algorithm B splits the identity space $\mathcal{ID}$ into two parts, $\mathcal{ID}_1$ and $\mathcal{ID}_2$, such that

	normal usk_{id}	**semi-funct.** usk_{id}
normal C_{id}	✓	✓
semi-funct. C_{id}	✓	✗

Table 5.1: Successful (✓) and unsuccessful (✗) decryption regarding normal and semi-functional user secret keys usk_{id} and ciphertexts C_{id}, respectively, for identity id.

- for identities $id \in \mathcal{ID}_1$ the reduction algorithm B is able to generate user secret keys
- for identities $id \in \mathcal{ID}_2$ the reduction algorithm B embeds an underlying complexity assumption, i.e. B can use these identities as challenge identity id^*, but is unable to generate user secret keys for id

This strategy has several drawbacks. Reductions in fully secure schemes partition the identity space according to the number of user secret key queries (poly in k), which leads to a large security loss. Furthermore, in such a reduction, the partitioning strategy is implemented in the public parameters, which makes them very large and impractical for some applications. For instance, the public parameters in Waters' [Wat05] efficient fully secure scheme in the standard model consists of $\mathbf{O}(k)$ group elements. Further, the strategy is not applicable for proving security for hierarchical IBE (HIBE) [HL02] or attribute-based encryption (ABE) [SW05] schemes, which offer more functionality. The last two disadvantages were removed using the following new approach due to Waters [Wat09].

Dual System Encryption. [Wat09] introduces a new technique, dubbed Dual System Encryption, for proving full security of IBE and HIBE schemes under simple assumptions. This technique yields IBE and HIBE schemes with public parameters, ciphertexts and user secret keys consisting of only a constant number of group elements.

In a Dual System Encryption system, both user secret keys and ciphertexts can either be *normal* or *semi-functional*. Both forms are indistinguishable from each other. A normal user secret key or ciphertext, respectively, is generated by using the key generation algorithm or encryption algorithm, respectively, of the scheme. A semi-functional user secret key or ciphertext instead is generated by using additional algorithms. These are not part of the original IBE or HIBE scheme and only used during the security proof. A semi-functional user secret key for an identity id is able to decrypt normal ciphertexts corresponding to id, but fail for semi-functional ciphertexts corresponding to id. Analogously, semi-functional ciphertexts are only decryptable using normal user secret keys corresponding to the same identity (see also Table 5.1).

In order to prove security of an IBE scheme, a sequence of games is defined as follows:

Game 0. Game 0 is the original security experiment of the IBE scheme, where all queried user secret keys usk_{id^j}, for $j = 1, \ldots, q = q(k)$, and the challenge ciphertext C_{id^*} are normal.

Game 1. Game 1 is defined as Game 0 apart from the fact that the challenge ciphertext is semi-functional.

Game 2.i. Game 2.i is defined as Game 1 except that all user secret keys usk_{id^j}, for $j = 1, \ldots, i$, are semi-functional and all user secret keys usk_{id^j}, for $j = i+1, \ldots, q$ are normal.

Game 3. Game 3 is defined as Game 2.q except that the the message encrypted in the challenge ciphertext is an independent uniform k-length bitstring.

In the security reduction [Wat09] shows that each game is indistinguishable from the next game under simple assumptions. Intuitively, for Game 0 and Game 1 they argue that no adversary can distinguish these games, since all user secret keys are still normal, hence, they can help to decrypt regardless whether the challenge ciphertext is normal or semi-functional. In Game 2.q all user secret keys are semi-functional, hence, it is no problem to change the challenge message to random as in Game 3, since no semi-functional user secret key is usable for decrypting a semi-functional challenge ciphertext. At the end, the challenge message is random and an adversary can only win by guessing.

One important point the authors had to deal with in their security proof was to ensure that it is not possible for the reduction algorithm to distinguish Game 2.i from Game 2.$i+1$ by itself, for instance by generating a semi-functional ciphertext $C_{id^{i+1}}$ for the corresponding user secret key $usk_{id^{i+1}}$ and test if decryption fails or not. They overcome this problem by embedding a polynomial $f(id) = a \cdot id + b$ of degree one, both in semi-functional user secret keys and semi-functional ciphertexts, such that the decryption algorithm will only work, i.e. decrypt correctly, if $f(id)$ is used only once as input, either for the semi-functional ciphertext or the semi-functional user secret key.

Since $q = q(k)$ hybrid games are necessary to change all user secret keys from normal to semi-functional, and each step requires a computational assumption, the security loss still depends on the number of user secret key queries. This can be improved using the following technique by Chen and Wee [CW13].

Nested Dual System Groups. Waters [Wat05] posed the important question, if it is possible to construct a fully secure IBE scheme with a tight security reduction under a standard assumption.

In [CW13] Chen and Wee constructed the first almost tight and fully secure IBE scheme under a standard assumption. Almost tight means that the security loss only depends on the security parameter k and is independent of the number of user secret key queries. In their approach they introduced an abstraction called Nested Dual System Groups (NDSG). NDSG can be seen as a variant of Dual System Groups (DSG) [CW14] which itself are based on the Dual System framework introduced by [Wat09]. Informally, NDSG consists of a triple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, satisfying a number of abstract properties, not further explained here.

[CW13] combines the Dual System Encryption technique with ideas of the analysis of the Naor-Reingold pseudorandom function [NR04]. In Dual System Encryption the semi-functional user secret keys are introduced one after the other, which results in q hybrid games to change all user secret keys from normal to semi-functional. The idea is to use the identities of the IBE scheme as input for random functions and to iterate over the input bits in the binary encoding of the identity, as done in the analysis of the Naor-Reingold PRF. Therefore, we consider identities $id = (id_1 \dots id_n) \in \mathcal{ID} = \{0,1\}^n, n = n(k)$ and random functions R_i, $i = 1, \dots, n$ for inputs $id|_i = id_1 \dots id_i$ (the i-bit prefix of id), more formally $R_i : \{0,1\}^i \to \mathbb{H}$, $R_i(id|_i) \to h_i$. Then we say, a challenge ciphertext or user secret key for an identity id is *semi-functional of type* i, if it contains additional randomness in form of $R_i(id|_i)$ (e.g. added by multiplying). Starting from 0-bit prefix to 1-bit prefix, 2-bit prefix ..., to the entire identity, the dependency on the input identity increases in every step. This property is called *nested hiding* (see also section 5.4).

This yields n hybrid games, described below. In Game 2.i the challenge ciphertext and each user secret key are semi-functional of type i, which means that their semi-functional component depends only on the first i bits of the corresponding identity id given as input to R_i. Hence, their entropy in the semi-functional component increases in every game from $R_i(id|_i)$ to $R_{i+1}(id|_{i+1})$, for $i = 1, \dots, n$. At the end $R_n(id^*)$, for the challenge identity id^*, is like a truly random value. Therefore, the challenge message is perfectly hidden and can be replaced by a random message.

Game 0. Game 0 is the original security experiment of the IBE scheme, where all queried user secret keys usk_{id^j}, for $j = 1, \dots, q = q(k)$, and the challenge ciphertext C_{id^*} are normal.

	s-f. type i usk_{id}		**s-f. type** $i+1$ usk_{id}	
	$id\|_i = id'\|_i$ $id\|_{i+1} \neq id'\|_{i+1}$	$id\|_i = id'\|_i$ $id\|_{i+1} = id'\|_{i+1}$	$id\|_i = id'\|_i$ $id\|_{i+1} \neq id'\|_{i+1}$	$id\|_i = id'\|_i$ $id\|_{i+1} = id'\|_{i+1}$
s-f. type i $C_{id'}$	✓	✓	✗	✗
s-f. type $i+1$ $C_{id'}$	✗	✗	✗	✓

Table 5.2: Successful (✓) and unsuccessful (✗) decryption regarding semi-functional type i and $i+1$ user secret keys usk_{id}, for identity id, and semi-functional type i and $i+1$ ciphertexts $C_{id'}$, for identity id'.

Game 1. Game 1 is defined as Game 0 apart from the fact that the challenge ciphertext is pseudo-normal (generated in a way, that enables gradual randomization).

Game 2.i Game 2.i is defined as Game 1 except that the challenge ciphertext and all user secret keys usk_{id^j}, for $j = 1, \ldots, q$, are semi-functional of type i.

Game 3. Game 3 is defined as Game 2.n except that the message encrypted in the challenge ciphertext is a uniform k-length bitstring.

Note, type i user secret keys usk_{id}, for an identity id, can decrypt type i ciphertexts $C_{id'}$, for an identity id', if $id|_i = id'|_i$, and fail otherwise (see also Table 5.2). If $id|_i = id'|_i$ both are multiplied with the same randomness $R_i(id|_i)$ and during decryption some kind of cancellation takes place. For different semi-functional types, e.g. semi-functional type i user secret key usk_{id} and semi-functional type $i+1$ ciphertext C_{id} (and vice versa) decryption fails even for the same identity id. Since the user secret key is multiplied by randomness $R_i(id|_i)$, which is different from the randomness $R_{i+1}(id|_{i+1})$ multiplied by the ciphertext, no cancellation happens and decryption fails.

Again, one problem arises with this strategy: The reduction algorithm B is also in a position to distinguish Game 2.i from Game 2.$i+1$ by itself (as mentioned in the Dual System Strategy), if B is able to create a semi-functional type i user secret key usk_{id^*} for the challenge identity id^* to test if decryption would succeed (i.e., C_{id^*} is semi-functional of type i) or fail (i.e., C_{id^*} is semi-functional of type $i+1$).

Hence, Chen and Wee move from the i-th to the $(i+1)$-th hybrid through a single reduction as follows: first, they guess the $(i+1)$-th bit id^*_{i+1} of the challenge identity id^*. Then, they set up things such that

- all user secret keys for identities id with $id_{i+1} = id^*_{i+1}$ (i.e., that coincide in the $(i+1)$-th bit with id^*) are of the same type as in the previous hybrid (i.e., carry only a blinding term $R_i(id|_i)$),
- all user secret keys for identities id with $id_i = 1 - id^*_i$ carry a blinding term of $R_i(id|_i) \cdot R'(id|_i)$. Depending on the input of the reduction, we have either that $R' = 1$ (such that the overall blinding term is $R(id|_i)$), or that R' is an independently random function (such that $R_i(id|_i) \cdot R'(id|_i) =: R_{i+1}(id|_{i+1})$).

The first property prevents the reduction B from testing by itself, since both semi-functional type i and type $i+1$ user secret keys for identities id, with $id|_{i+1} = id^*|_{i+1}$, are the same, which

	s-f. type i usk_{id}		**s-f. type** $i+1$ usk_{id}	
	$id\|_i = id'\|_i$ $id\|_{i+1} \neq id'\|_{i+1}$	$id\|_i = id'\|_i$ $id\|_{i+1} = id'\|_{i+1}$	$id\|_i = id'\|_i$ $id\|_{i+1} \neq id'\|_{i+1}$	$id\|_i = id'\|_i$ $id\|_{i+1} = id'\|_{i+1}$
s-f. type $(\wedge, i)$ $C_{id'}$	✓	✓	✗	✗
s-f. type $(\sim, i)$ $C_{id'}$	✓	✓	✗	✗
s-f. type $(\wedge, i+1)$ $C_{id'}$	✗	✗	✗	✓
s-f. type $(\sim, i+1)$ $C_{id'}$	✗	✗	✗	✓

Table 5.3: Successful (✓) and unsuccessful (✗) decryption regarding semi-functional type i and $i+1$ user secret keys usk_{id}, for identity id, and semi-functional type $(\wedge, i)$, $(\sim, i)$, $(\wedge, i+1)$ and $(\sim, i+1)$ ciphertexts $C_{id'}$, for identity id'.

means that both types successfully decrypt a semi-functional type i and type $i+1$ challenge ciphertext C_{id^*}.

Depending on whether or not $R' = 1$, this setup simulates the i-th or the $(i+1)$-th hybrid.

Problem of this Approach in the Multi-Instance, Multi-Ciphertext Setting. In their $(i+1)$-th game hop, only challenge ciphertexts for identities with the same $(i+1)$-th bit can be generated. In the multi-instance, multi-ciphertext setting we have many different challenge ciphertexts for corresponding identities. Since the identities are not equal they differ in at least one bit. Thus, their approach cannot in any obvious way be extended to multiple challenge ciphertexts for different identities. For similar reasons, a generalization to multiple instances of the scheme fails. To overcome this problem, we introduce Extended Nested Dual System Groups (ENDSG), which are formally described in section 5.4. The idea of the strategy is briefly described here.

Proof Strategy for Multi-Instance, Multi-Ciphertext IBE Schemes. Informally, extended nested dual system groups (ENDSG) are a tuple of groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, \widehat{\mathbb{G}}, \widetilde{\mathbb{G}}, \widehat{\mathbb{H}}, \widetilde{\mathbb{H}})$ and a bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, where $\mathbb{G}, \mathbb{H}$ and $\mathbb{G}_T$ are composite-order groups of the same order with proper subgroups $\widehat{\mathbb{G}}, \widetilde{\mathbb{G}} \subset \mathbb{G}$ and $\widehat{\mathbb{H}}, \widetilde{\mathbb{H}} \subset \mathbb{H}$ of different order.

Similar to NDSG, we use random functions R_i to increase the entropy in our semi-functional objects. But, in comparison, we distribute the blinding terms $R_i(id|_i)$ used in NDSG in two different subgroups, i.e., we use two different random functions $\widehat{R}_i(id|_i)$ and $\widetilde{R}_i(id|_i)$ instead, which map in $\widehat{\mathbb{H}}$ and $\widetilde{\mathbb{H}}$, respectively. In our case, we have semi-functional type-$(\wedge, i)$ and type-$(\sim, i)$ ciphertexts, which contain either $\widehat{\mathbb{G}}$-elements and $\widehat{R}_i(id|_i)$ or $\widetilde{\mathbb{G}}$-elements and $\widetilde{R}_i(id|_i)$, and semi-functional type i user secret keys, which contain both $\widehat{R}_i(id|_i)$ and $\widetilde{R}_i(id|_i)$ and no additional $\widehat{\mathbb{G}}$- or $\widetilde{\mathbb{G}}$-elements. For decryption rules see Table 5.3.

In order to move from the $(i-1)$-th to the i-th hybrid, we thus follow a different strategy that involves three reductions. We only describe these three games in the following, since the other games are similar to the games in NDSG adapted to the multi-instance, multi-ciphertext setting (for more details see section 5.4.1):

Game 2.i.0. Game 2.i is defined as Game 1 except that all user secret keys are semi-functional of type $i-1$ and all challenge ciphertexts are semi-functional of type $(\wedge, i-1)$.

Game 2.i.1. Game 2.i.1 is defined as Game 2.i.0 except that if the i-th bit of a challenge identity

is 0, then the corresponding challenge ciphertext is semi-functional of type $(\wedge, i-1)$, otherwise if the bit is 1 then it is semi-functional of type $(\sim, i-1)$. (Note, that all user secret keys are still semi-functional of type $i-1$.)

Game 2.i.2. Game $2.i.2$ is defined as Game $2.i.1$ except that the challenge ciphertexts are semi-functional of type $(\cdot, i)$ (where $\cdot$ can be $\wedge$ or $\sim$ as defined in game $2.i.1$ depending on the i-th bit of the challenge identity) and all user secret keys are semi-functional of type i.

We take advantage of the property of the bilinear map, that elements from $\wedge-$ and $\sim -$ subgroups behave as follows under the bilinear map e: $e(\widehat{g},\widetilde{h}) = e(\widetilde{g},\widehat{h}) = 1$ (i.e., cancellation) and $e(\widehat{g},\widehat{h}) = e(\widetilde{g},\widetilde{h}) \neq 1$ (i.e., reaction), for all $\widehat{g} \in \widehat{\mathbb{G}}, \widetilde{g} \in \widetilde{\mathbb{G}}, \widehat{h} \in \widehat{\mathbb{H}}, \widetilde{h} \in \widetilde{\mathbb{H}}$. In other words, cancellation is needed for successful decryption and if there is any reaction, decryption fails. During the reduction this enables to embed a computational challenge in the form of $(\widehat{R}', \widetilde{R}')$ in the user secret keys usk_{id}. Depending on the ith-bit id_i of the identity for a corresponding user secret key, we embed the challenge either in

$$\widehat{R}_i(id|_i) = \widehat{R}_{i-1}(id|_{i-1}) \cdot \widehat{R}' \text{ and set } \widetilde{R}_i(id|_i) := \widetilde{R}_{i-1}(id|_{i-1}), \text{ if } id_i = 1,$$

or

$$\widetilde{R}_i(id|_i) = \widetilde{R}_{i-1}(id|_{i-1}) \cdot \widetilde{R}' \text{ and set } \widehat{R}_i(id|_i) := \widehat{R}_{i-1}(id|_{i-1}), \text{ if } id_i = 0.$$

Again, depending on $\widehat{R}' = \widetilde{R}' = 1$ or not, this is exactly Game $2.i.1$ or $2.i.2$.

Note again, testing with the challenge ciphertexts (or any other ciphertext) is not possible. We have semi-functional type $(\wedge, i)$ ciphertexts if $id_i^* = 0$, but for all user secret keys usk_{id} with $id_i = 0$, we embedded an additional term $\widetilde{R}'$ in the user secret key. Since the $\wedge-$ and $\sim -$ components cancel each other out, decryption succeeds no matter if the key is semi-functional of type $i-1$ or i (for identities with the same i-bit prefix). Hence, a "reaction" (resulting in a failing decryption), that would be helpful to distinguish the games, is not possible for these challenge ciphertexts and user secret keys. Analogously for $id_i^* = 1$.

In other words, we shift the challenge ciphertexts according to their ith-bit either in the $\wedge-$semi-functional space or in the $\sim -$semi-functional space. Then, we apply Chen and Wee's proof strategy in both subspaces, seperately.

From a conceptual perspective, it might also be interesting to note that none of our reductions needs to *guess*, e.g., an identity bit.

5.4 Extended Nested Dual System Groups (ENDSG)

In the following, based on NDSGs, we construct a new notion we call extended nested dual system groups.

A Variant of Nested Dual System Groups. We introduce a variant of Chen and Wee's nested dual system groups (NDSG) [CW13], dubbed extended nested dual system groups (ENDSG) defined in Definition 39. Mainly, we re-use and extend the notions from [CW13].

Group Generation. Further, let $\mathsf{Grp}(1^k, n')$ be a group generation algorithm as defined in Definition 3, i.e.

$$(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, (g_{p_1}, \ldots, g_{p_{n'}}), (h_{p_1}, \ldots, h_{p_{n'}}), g, h, e) \leftarrow \mathsf{Grp}(1^k, n')$$

for a pairing $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, for composite-order groups $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$, all of known group order $N = p_1 \cdots p_{n'}$, and generators g and h. Further, $(g_{p_i})_i$ and $(h_{p_i})_i$ are generators of the (proper) subgroups $\mathbb{G}_{p_i} \subset \mathbb{G}$ and $\mathbb{H}_{p_i} \subset \mathbb{H}$ of order p_i.

Definition 39 (ENDSG)**.** An *extended nested dual system group* $\mathsf{ENDSG} = (\mathsf{SampP}, \mathsf{SampG}, \mathsf{SampH}, \widehat{\mathsf{SampG}}, \widetilde{\mathsf{SampG}})$ consists of five PPT algorithms:

Parameter Sampling. $\mathsf{SampP}(1^k, n)$, on input a security parameter 1^k and parameter $n \in \mathbb{N}$, samples

$$(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, (g_{p_1}, \ldots, g_{p_{n'}}), (h_{p_1}, \ldots, h_{p_{n'}}), g, h, e) \leftarrow \mathsf{Grp}(1^k, n'),$$

for a constant integer n' depending on n and determined by SampP, and outputs public parameters

$$pp = (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, g, h, e, m, n, \mathsf{pars})$$

and secret parameters

$$sp = (\widehat{h}, \widetilde{h}, \widehat{\mathsf{pars}}, \widetilde{\mathsf{pars}}),$$

where $m : \mathbb{H} \to \mathbb{G}_T$ is a linear map, $\widehat{h}, \widetilde{h}$ are nontrivial $\mathbb{H}$-elements, and $\mathsf{pars}, \widehat{\mathsf{pars}}, \widetilde{\mathsf{pars}}$ may contain arbitrary additional information used by $\mathsf{SampG}, \mathsf{SampH}$, and $\widehat{\mathsf{SampG}}$ and $\widetilde{\mathsf{SampG}}$.

$\mathbb{G}$-group Sampling. $\mathsf{SampG}(pp)$, on input public parameters pp, outputs $\mathbf{g} = (g_0, \ldots, g_n) \in \mathbb{G}^{n+1}$.

$\mathbb{H}$-group Sampling. $\mathsf{SampH}(pp)$, on input public parameters pp, outputs $\mathbf{h} = (h_0, \ldots, h_n) \in \mathbb{H}^{n+1}$.

Semi-Functional $\mathbb{G}$-group Sampling 1. $\widehat{\mathsf{SampG}}(pp, sp)$, on input public parameters pp and secret parameters sp, outputs $\widehat{\mathbf{g}} = (\widehat{g}_0, \ldots, \widehat{g}_n) \in \mathbb{G}^{n+1}$.

Semi-Functional $\mathbb{G}$-group Sampling 2. $\widetilde{\mathsf{SampG}}(pp, sp)$, on input public parameters pp and secret parameters sp, outputs $\widetilde{\mathbf{g}} = (\widetilde{g}_0, \ldots, \widetilde{g}_n) \in \mathbb{G}^{n+1}$.

Correctness of ENDSG. For *correctness*, we require for all $k \in \mathbb{N}$, for all integers $n = n(k) > 1$, for all pp, where pp is the first output of $\mathsf{SampP}(1^k, n)$, the following properties:

Associativity. For all $(g_0, \ldots, g_n) \leftarrow \mathsf{SampG}(pp)$ and for all $(h_0, \ldots, h_n) \leftarrow \mathsf{SampH}(pp)$, we have $e(g_0, h_i) = e(g_i, h_0)$, for all $i \in [n]$.

Projective. For all $s \in \mathbb{Z}_N^*$, for g_0 which is the first output of $\mathsf{SampG}(pp; s)$, for all $h \in \mathbb{H}$, we have $m(h)^s = e(g_0, h)$.

Security of ENDSG. For security, we require for all $k \in \mathbb{N}$, for all integers $n = n(k) > 1$, for all $(pp, sp) \leftarrow \mathsf{SampP}(1^k, n)$, the following properties:

Orthogonality. For m specified in pp, for $\widehat{h}, \widetilde{h}$ specified in sp, we have

$$m(\widehat{h}) = m(\widetilde{h}) = 1.$$

For g_0, $\widehat{g}_0$, and $\widetilde{g}_0$ that are the first outputs of $\mathsf{SampG}(pp)$, $\widehat{\mathsf{SampG}}(pp, sp)$, and $\widetilde{\mathsf{SampG}}(pp, sp)$, respectively, we have that

$$e(g_0, \widehat{h}) = 1, e(g_0, \widetilde{h}) = 1, e(\widehat{g}_0, \widetilde{h}) = 1, \text{ and } e(\widetilde{g}_0, \widehat{h}) = 1.$$

$\mathbb{G}$- and $\mathbb{H}$-subgroups. The outputs of SampG, $\widehat{\mathsf{SampG}}$, and $\widetilde{\mathsf{SampG}}$ are distributed uniformly over the generators of different nontrivial subgroups of $\mathbb{G}^{n+1}$ (that only depend on pp) of coprime order, respectively, while the output of SampH is uniformly distributed over the generators of a nontrivial subgroup of $\mathbb{H}^{n+1}$ (that only depends on pp).

Non-degeneracy. For $\widehat{h}$ specified in sp and for $\widehat{g}_0$ which is the first output of $\widehat{\mathsf{SampG}}(pp, sp)$, it holds that $e(\widehat{g}_0, \widehat{h})$ is uniformly distributed over the generators of a nontrivial subgroup of $\mathbb{G}_T$ (that only depends on pp). Similarly, $e(\widetilde{g}_0, \widetilde{h})$ is uniformly distributed over the generators of a nontrivial subgroup of $\mathbb{G}_T$ (that only depends on pp), where $\widetilde{h}$ is specified in sp and $\widetilde{g}_0$ is the first output of $\widetilde{\mathsf{SampG}}(pp, sp)$.

Left-subgroup indistinguishability 1 (LS1). For any PPT adversary D, we have that the function

$$\mathsf{Adv}^{\mathrm{ls1}}_{\mathsf{ENDSG},\mathsf{Grp},D}(k,n) := |\Pr[D(pp,\mathbf{g})=1] - \Pr[D(pp,\mathbf{g}\widehat{\mathbf{g}})=1]|$$

is negligible in k, where $\mathbf{g} \leftarrow \mathsf{SampG}(pp)$, $\widehat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(pp, sp)$.

Left-subgroup indistinguishability 2 (LS2). For any PPT adversary D, we have that the function

$$\mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG},\mathsf{Grp},D}(k,n) := |\Pr\left[D(pp,\widehat{h}\widetilde{h},\mathbf{g}'\widehat{\mathbf{g}}',\mathbf{g}\widehat{\mathbf{g}})=1\right] - \Pr\left[D(pp,\widehat{h}\widetilde{h},\mathbf{g}'\widehat{\mathbf{g}}',\mathbf{g}\widetilde{\mathbf{g}})=1\right]|$$

is negligible in k, where $\mathbf{g},\mathbf{g}' \leftarrow \mathsf{SampG}(pp)$, $\widehat{\mathbf{g}},\widehat{\mathbf{g}}' \leftarrow \widehat{\mathsf{SampG}}(pp, sp)$, $\widetilde{\mathbf{g}} \leftarrow \widetilde{\mathsf{SampG}}(pp, sp)$, for $\widehat{h}$ and $\widetilde{h}$ specified in sp.

Nested-hiding indistinguishability (NH). For any PPT adversary D, for all $q' = q'(k)$, the function

$$\begin{aligned}\mathsf{Adv}^{\mathrm{nh}}_{\mathsf{ENDSG},\mathsf{Grp},D}(k,n,q') := \max_{i\in[\lfloor \frac{n}{2} \rfloor]} \Big(&|\Pr\left[D(pp,\widehat{h},\widetilde{h},\widehat{\mathbf{g}}_{-(2i-1)},\widetilde{\mathbf{g}}_{-2i},(\mathbf{h}_1,\ldots,\mathbf{h}_{q'}))=1\right] \\ &- \Pr\left[D(pp,\widehat{h},\widetilde{h},\widehat{\mathbf{g}}_{-(2i-1)},\widetilde{\mathbf{g}}_{-2i},(\mathbf{h'}_1,\ldots,\mathbf{h'}_{q'}))=1\right]|\Big),\end{aligned}$$

is negligible in k, where $\widehat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(pp, sp)$, $\widetilde{\mathbf{g}} \leftarrow \widetilde{\mathsf{SampG}}(pp, sp)$, and

$$\mathbf{h}_{i'} := (h_{i',0},\ldots,h_{i',n}) \leftarrow \mathsf{SampH}(pp),$$

$$\mathbf{h}'_{i'} := (h_{i',0},\ldots,h_{i',2i-1}\cdot(\widehat{h})^{\widehat{\gamma}_{i'}}, h_{i',2i}\cdot(\widetilde{h})^{\widetilde{\gamma}_{i'}},\ldots,h_{i',n}),$$

for $i \in \{0,\ldots,n\}$, for $\widehat{h},\widetilde{h}$ specified in sp, for $\widehat{\gamma}_{i'},\widetilde{\gamma}_{i'} \leftarrow \mathbb{Z}^*_{\mathrm{ord}(\mathbb{H})}$, and for all $i' \in [q']$.

(Informal) Comparison of NDSGs and ENDSGs. Loosely speaking, in contrast to the NDSGs from [CW13], ENDSGs have a second semi-functional $\mathbb{G}$-group sampling algorithm $\widetilde{\mathsf{SampG}}$ as well as a second nontrivial $\mathbb{H}$-element $\widetilde{h}$ in sp. Further, we omit the SampGT-algorithm. Concerning the ENDSG properties, we extend the NDSG properties and assumptions appropriately and introduce one additional assumption (i.e., LS2).

5.4.1 Generic Construction

We emphasize here, that this section is not part of this thesis and extensively discussed in [Str15]. It is reproduced here for a better understanding of the result in our next section, where we construct an instantiation of this scheme.

A variant of the IBE of [CW13]. [HKS15] presents a variant of Chen and Wee's IBE scheme [CW13]. As a basic building block an extended nested dual system group $\mathsf{ENDSG} = (\mathsf{SampP},\mathsf{SampG},\mathsf{SampH},\widehat{\mathsf{SampG}},\widetilde{\mathsf{SampG}})$ as described in section 5.4 is used.

Besides, for groups $\mathbb{G}_T$ (defined below), let $\mathcal{UH}$ be a family of universal hash functions $\mathsf{H}: \mathbb{G}_T \to \{0,1\}^k$ such that for any nontrivial subgroup $\mathbb{G}'_T \subset \mathbb{G}_T$, and for $\mathsf{H} \leftarrow \mathcal{UH}$, $X \leftarrow \mathbb{G}'_T$, and $U \leftarrow \{0,1\}^k$, it holds $\mathsf{SD}((\mathsf{H},\mathsf{H}(X));(\mathsf{H},U)) = \mathbf{O}(2^{-k})$.

Let $\mathsf{IBE} = (\mathsf{Par},\mathsf{Gen},\mathsf{Ext},\mathsf{Enc},\mathsf{Dec})$ with identity space $\mathcal{ID} = \{0,1\}^n$, for $n = n(k)$ a priori, and message space $\mathcal{M}_k = \{0,1\}^k$ be defined as follows:

Parameter Generation. $\mathsf{Par}(1^k, n)$, on input a security parameter 1^k and a parameter $n \in \mathbb{N}$, samples $(pp', sp') \leftarrow \mathsf{SampP}(1^k, 2n)$

with

$$pp' = (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, g, h, e, m, 2n, \mathsf{pars})$$

and

$$sp' = (\widehat{h}, \widetilde{h}, \widehat{\mathsf{pars}}, \widetilde{\mathsf{pars}}).$$

Further, $\mathsf{Par}(1^k, n)$ samples a universal hash function

$$\mathsf{H} \leftarrow \mathcal{UH}.$$

Finally, $\mathsf{Par}(1^k, n)$ outputs the public and secret parameters (pp, sp), where

$$pp := (pp', \mathsf{H}) \text{ and } sp := sp'.$$

Key Generation. $\mathsf{Gen}(pp, sp)$, on input public parameters pp and secret parameters sp, samples $msk \leftarrow \mathbb{H}$, and outputs a master public key and master secret key

$$(mpk, msk) := ((pp, m(msk)), msk).$$

Secret Key Extraction. $\mathsf{Ext}(msk, id)$, on input a master secret key $msk \in \mathbb{H}$ and an identity $id = (id_1 \dots id_n) \in \mathcal{ID}$, samples $(h_0, \dots, h_{2n}) \leftarrow \mathsf{SampH}(pp)$ and outputs a user secret key

$$usk_{id} := (h_0, msk \cdot \prod_{i=1}^{n} h_{2i-id_i}) =: (K_0, K_1).$$

Encryption. $\mathsf{Enc}(mpk, id, M)$, on input a master public key $mpk = (pp, m(msk))$, an identity $id = (id_1 \dots id_n) \in \mathcal{ID}$, and a message $M \in \mathcal{M}_k$, computes

$$(g_0, \dots, g_{2n}) := \mathsf{SampG}(pp; s), \text{ for } s \leftarrow \mathbb{Z}_N^*,$$

and

$$g_T := m(msk)^s \overset{(*)}{=} e(g_0, msk),$$

where (*) holds due to ENDSG's projective property, and outputs a ciphertext

$$C_{id} := (g_0, \prod_{i=1}^{n} g_{2i-id_i}, \mathsf{H}(g_T) \oplus M) =: (C_0, C_1, C_2).$$

Decryption. $\mathsf{Dec}(usk_{id}, C_{id'})$, on input a user secret key $usk_{id} =: (K_0, K_1)$ and a ciphertext $C_{id'} =: (C_0, C_1, C_2)$, outputs

$$M := \mathsf{H}\left(\frac{e(C_0, K_1)}{e(C_1, K_0)}\right) \oplus C_2.$$

Correctness of IBE. For $id = id'$ it holds

$$\begin{aligned}
\mathsf{H}\left(\frac{e(C_0, K_1)}{e(C_1, K_0)}\right) \oplus C_2 &= \mathsf{H}\left(\frac{e(g_0, msk \cdot \prod_{i=1}^{n} h_{2i-id_i})}{e(\prod_{i=1}^{n} g_{2i-id'_i}, h_0)}\right) \oplus (\mathsf{H}(g_T) \oplus M) \\
&= \mathsf{H}\left(\frac{e(g_0, msk) \prod_{i=1}^{n} e(g_0, h_{2i-id_i})}{\prod_{i=1}^{n} e(g_{2i-id'_i}, h_0)}\right) \oplus \mathsf{H}(g_T) \oplus M \\
&\overset{(*)}{=} \mathsf{H}(e(g_0, msk)) \oplus \mathsf{H}(g_T) \oplus M \overset{(*)}{=} \mathsf{H}(g_T) \oplus \mathsf{H}(g_T) \oplus M \\
&= M,
\end{aligned}$$

$(*)$ holds due to ENDSG's associativity and projective properties.

(μ, q)-IBE-IND-CPA Security of IBE. The high-level proof strategy is based on the IBE-IND-CPA proof strategy of [CW13], but deviate on the low level. In order to provide an overview on the proof strategy, auxiliary secret-key extraction $\overline{\mathsf{Ext}}$ and auxiliary encryption $\overline{\mathsf{Enc}}$, random functions $\widehat{\mathsf{R}}_{j,i}$ and $\widetilde{\mathsf{R}}_{j,i}$, pseudo-normal ciphertexts, semi-functional type-$(\cdot, i)$ ciphertexts, and semi-functional type-i user secret keys similarly to [CW13] are defined:

Auxiliary Secret Key Extraction. $\overline{\mathsf{Ext}}(pp, msk, id; \mathbf{h})$, on input public parameters pp, a master secret key msk, an identity $id = id_1 \dots id_n \in \mathcal{ID}$, and $\mathbf{h} = (h_0, \dots, h_{2n}) \in \mathbb{H}^{2n+1}$, outputs a user secret key

$$usk_{id} := (h_0, msk \cdot \prod_{i=1}^{n} h_{2i-id_i}).$$

Auxiliary Encryption Function. $\overline{\mathsf{Enc}}(pp, id, M; msk, \mathbf{g})$, on input a public parameter pp, an identity $id = id_1 \dots id_n \in \mathcal{ID}$, a message $M \in \mathcal{M}_k$, a master secret key msk, and $\mathbf{g} = (g_0, \dots, g_{2n}) \in (\mathbb{G})^{2n+1}$, outputs a ciphertext

$$C_{id} := (g_0, \prod_{i=1}^{n} g_{2i-id_i}, \mathsf{H}(e(g_0, msk)) \oplus M).$$

Random Function Families. Let

$$id|_i := id_1 \dots id_i \in \{0,1\}^i =: \mathcal{ID}|_i$$

be the i-bit prefix of an identity id. For an instance $j \in [\mu]$ and $i \in [n] \cup \{0\}$, consider the following independent and truly random functions

$$\widehat{\mathsf{R}}_{j,i} : \mathcal{ID}|_i \to \mathbb{H},\ id|_i \mapsto (\widehat{h})^{\widehat{\gamma}_{j,i}(id|_i)}$$

and

$$\widetilde{\mathsf{R}}_{j,i} : \mathcal{ID}|_i \to \mathbb{H},\ id|_i \mapsto (\widetilde{h})^{\widetilde{\gamma}_{j,i}(id|_i)},$$

where

$$\widehat{\gamma}_{j,i} : \mathcal{ID}|_i \to \mathbb{Z}^*_{\mathrm{ord}(\mathbb{H})}, id|_i \mapsto \widehat{\gamma}_{j,id|_i}$$

and

$$\widetilde{\gamma}_{j,i} : \mathcal{ID}|_i \to \mathbb{Z}^*_{\mathrm{ord}(\mathbb{H})}, id|_i \mapsto \widetilde{\gamma}_{j,id|_i}.$$

Pseudo-Normal Ciphertexts. Pseudo-normal ciphertexts are generated as

$$\begin{aligned} C_{id} &:= \overline{\mathsf{Enc}}(pp, id, M; msk, \mathbf{g}\widehat{\mathbf{g}}) \\ &= (g_0\widehat{g}_0, \prod_{i=1}^{n} g_{2i-id_i}\widehat{g}_{2i-id_i}, \mathsf{H}(e(g_0\widehat{g}_0, msk)) \oplus M), \end{aligned}$$

for uniform $\mathbf{g} = (g_0, \dots, g_{2n}) \leftarrow \mathsf{SampG}(pp)$ and $\widehat{\mathbf{g}} = (\widehat{g}_0, \dots, \widehat{g}_{2n}) \leftarrow \widehat{\mathsf{SampG}}(pp, sp)$. (Hence, pseudo-normal ciphertexts have $\mathbb{G}$-components sampled from $\widehat{\mathsf{SampG}}$ and thus msk is needed as additional input to replace $g_T = m(msk)^s = e(g_0, msk)$ by $e(g_0\widehat{g}_0, msk)$).

Semi-Functional Type-$(\wedge, i)$ and Type-$(\sim, i)$ Ciphertexts. Let $\widehat{\mathsf{R}}_{j,i}$ and $\widetilde{\mathsf{R}}_{j,i}$ be random functions as defined above. Semi-functional ciphertexts of type $(\wedge, i)$ are generated as

$$\begin{aligned} \widehat{C}_{id} &:= \overline{\mathsf{Enc}}(pp, id, M; msk \cdot \widehat{\mathsf{R}}_{j,i}(id|_i) \cdot \widetilde{\mathsf{R}}_{j,i}(id|_i), \mathbf{g}\widehat{\mathbf{g}}) \\ &\overset{(1)}{=} (g_0\widehat{g}_0, \prod_{i=1}^{n} g_{2i-id_i}\widehat{g}_{2i-id_i}, \mathsf{H}(e(g_0\widehat{g}_0, msk \cdot \widehat{\mathsf{R}}_{j,i}(id|_i))) \oplus M) \end{aligned}$$

while semi-functional ciphertexts of type $(\sim, i)$ are generated as

$$\begin{aligned}\widetilde{C}_{id} &:= \overline{\mathsf{Enc}}(pp, id, M; msk \cdot \widehat{\mathsf{R}}_{j,i}(id|_i) \cdot \widetilde{\mathsf{R}}_{j,i}(id|_i), \mathbf{g}\widetilde{\mathbf{g}}) \\ &\overset{(2)}{=} (g_0\widetilde{g}_0, \prod_{i=1}^{n} g_{2i-id_i}\widetilde{g}_{2i-id_i}, \mathsf{H}(e(g_0\widetilde{g}_0, msk \cdot \widetilde{\mathsf{R}}_{j,i}(id|_i))) \oplus M),\end{aligned}$$

where $\mathbf{g} = (g_0, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(pp)$, $\widehat{\mathbf{g}} = (\widehat{g}_0, \ldots, \widehat{g}_{2n}) \leftarrow \widehat{\mathsf{SampG}}(pp)$, and $\widetilde{\mathbf{g}} = (\widetilde{g}_0, \ldots, \widetilde{g}_{2n}) \leftarrow \widetilde{\mathsf{SampG}}(pp)$, while (1) and (2) hold due to ENDSG's properties.

Semi-Functional Type-i User Secret Keys. Let $\widehat{\mathsf{R}}_{j,i}$ and $\widetilde{\mathsf{R}}_{j,i}$ be defined as above. For $\mathbf{h} = (h_0, \ldots, h_{2n}) \leftarrow \mathsf{SampH}(pp)$, semi-functional type-$i$ user secret keys are generated as

$$\begin{aligned}usk_{id} &:= \overline{\mathsf{Ext}}(pp, msk \cdot \widehat{\mathsf{R}}_{j,i}(id|_i) \cdot \widetilde{\mathsf{R}}_{j,i}(id|_i), id; \mathbf{h}) \\ &= (h_0, msk \cdot \widehat{\mathsf{R}}_{j,i}(id|_i) \cdot \widetilde{\mathsf{R}}_{j,i}(id|_i) \cdot \prod_{i=1}^{n} h_{2i-id_i}).\end{aligned}$$

Theorem 12. If ENDSG is an extended nested dual system group system as defined in section 5.4 and H is a universal hash function, then IBE defined as above is weakly (μ, q)-IBE-IND-CPA-secure. Concretely, for any weak PPT adversary A with at most $q' = q'(k)$ key extraction queries per instance and running time t in the (μ, q)-IBE-IND-CPA security experiment (μ instances, q challenge ciphertexts) with IBE, there are distinguishers D_1 on LS1, D_2 on LS2, and D_3 on NH with running times $t'_1 \approx t'_2 \approx t'_3 \approx t + \mathbf{O}(\mu n k^c(q + q'))$, respectively, for some constant $c \in \mathbb{N}$, with

$$\begin{aligned}\mathsf{Adv}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k, n) \leq\ & \mathsf{Adv}^{\mathrm{ls1}}_{\mathsf{ENDSG},\mathsf{Grp},D_1}(k, 2n) + 2n \cdot \mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG},\mathsf{Grp},D_2}(k, 2n) \\ & + n \cdot \mathsf{Adv}^{\mathrm{nh}}_{\mathsf{ENDSG},\mathsf{Grp},D_3}(k, 2n, \mu q') + \mu q \cdot \mathbf{O}(2^{-k}),\end{aligned} \tag{5.1}$$

for a group generation algorithm Grp defined as in Definition 3.

The full proof can be found in [HKS15] and also in [Str15] in more detail. We only outline the idea of the proof to clarify where the security properties of ENDSG are used.

Idea of the Proof Strategy. [HKS15] shows the (μ, q)-IBE-IND-CPA security of IBE for any weak PPT adversary A in a sequence of games where they successively change the games until they arrive at a game where A has only negligible advantage (i.e., success probability of $1/2$) in the sense of (μ, q)-IBE-IND-CPA. In Table 5.4 an overview how the challenge ciphertexts and user secret keys are generated is given.

Game 0. Game 0 is the (μ, q)-IBE-IND-CPA experiment as defined above.

Game 1. Game 1 is defined as Game 0 apart from the fact that all challenge ciphertexts are pseudo-normal.

Game 2.i.0. Game 2.i.0 is defined as Game 1 except that all user secret keys are semi-functional of type $(i - 1)$ and all challenge ciphertexts are semi-functional of type-$(\wedge, i - 1)$, for all $i \in [n]$.

Game 2.i.1. Game 2.i.1 is defined as Game 2.i.0 except that if the i-th bit of a challenge identity is 1, then the corresponding challenge ciphertext is semi-functional of type $(\sim, i - 1)$. (Otherwise, if the i-th bit of a challenge identity is 0, then the corresponding challenge ciphertext is semi-functional of type $(\wedge, i - 1)$.)

Game 2.i.2. Game 2.i.2 is defined as Game 2.i.1 except that the challenge ciphertexts are semi-functional of type $(\cdot, i)$ (where $\cdot$ can be $\wedge$ or $\sim$ as defined in Game 2.i.1, i.e., depending on the i-th challenge identity bit) and all the user secret keys are semi-functional of type i.

Game 3. Game 3 is defined as Game $2.n.2$ except that the challenge ciphertexts are semi-functional of type $(\wedge, n)$ (the user secret keys are semi-functional of type n).

Game 4. Game 4 is defined as Game 3 except that the messages encrypted in the challenge ciphertexts are uniform k-length bitstrings.

Further, Table 5.4 indicates which assumption or property, respectively, is used to show the indistinguishability of the corresponding games (marked with a label on the arrows). The corresponding proofs can be found in [HKS15] and [Str15] and are not part of this thesis. Taking them together, this shows Theorem 12.

	Game	Challenge ciphertexts for $id^*_{j,i'}$	User secret keys for id
LS1	G. 0	$\mathsf{Enc}(mpk_j, id^*_{j,i'}, M^*_{j,i',b})$	$\mathsf{Ext}(msk_j, id)$
orthog.	G. 1	$\underline{\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j, \mathbf{g}\widehat{\mathbf{g}})}$	$\underline{\overline{\mathsf{Ext}}(pp, msk_j, id; \mathbf{h})}$
LS2	G. $2.i.0$	$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,i-1}(id^*_{j,i'}\|_{i-1})}, \mathbf{g}\widehat{\mathbf{g}})$	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,i-1}(id\|_{i-1}) \cdot \widetilde{\mathsf{R}}_{j,i-1}(id\|_{i-1})}, id; \mathbf{h})$
NH	G. $2.i.1$	if $id^*_{j,i',i} = 0$:	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}_{j,i-1}(id\|_{i-1}) \cdot \widetilde{\mathsf{R}}_{j,i-1}(id\|_{i-1}), id; \mathbf{h})$
		$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \widehat{\mathsf{R}}_{j,i-1}(id^*_{j,i'}\|_{i-1}), \mathbf{g}\widehat{\mathbf{g}})$	
		if $id^*_{j,i',i} = 1$:	
		$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widetilde{\mathsf{R}}_{j,i-1}(id^*_{j,i'}\|_{i-1})}, \underline{\mathbf{g}\widetilde{\mathbf{g}}})$	
$i = n$: LS2	G. $2.i.2$	if $id^*_{j,i',i} = 0$:	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,i}(id\|_i) \cdot \widetilde{\mathsf{R}}_{j,i}(id\|_i)}, id; \mathbf{h})$
		$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,i}(id^*_{j,i'}\|_i)}, \mathbf{g}\widehat{\mathbf{g}})$	
		if $id^*_{j,i',i} = 1$:	
		$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widetilde{\mathsf{R}}_{j,i}(id^*_{j,i'}\|_i)}, \mathbf{g}\widetilde{\mathbf{g}})$	
stat.ind.	G. 3	$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,n}(id^*_{j,i'})}, \mathbf{g}\widehat{\mathbf{g}})$	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,n}(id) \cdot \widetilde{\mathsf{R}}_{j,n}(id)}, id; \mathbf{h})$
	G. 4	$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, \underline{R_{j,i'}}; msk_j \cdot \widehat{\mathsf{R}}_{j,n}(id^*_{j,i'}), \mathbf{g}\widehat{\mathbf{g}})$	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}_{j,n}(id) \cdot \widetilde{\mathsf{R}}_{j,n}(id), id; \mathbf{h})$

Table 5.4: Instance-j challenge ciphertexts for challenge identity $id^*_{j,i'}$, for $\mathbf{g} \leftarrow \mathsf{SampG}(pp)$, for $\widehat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(pp, sp)$, for $\widetilde{\mathbf{g}} \leftarrow \widetilde{\mathsf{SampG}}(pp, sp)$, for $R_{j,i'} \leftarrow \{0,1\}^k$, and for instance-$j$ user secret keys for identity id, for $\mathbf{h} \leftarrow \mathsf{SampH}(pp)$, for all $(j, i', i) \in [\mu] \times [q] \times [n]$. The differences between games are given by underlining.

From Weak to Full (μ, q)-IBE-IND-CPA Security. The theorem above shows only weak security: One must assume that the adversary A never asks for encryptions under the same challenge identity and for the same scheme instance twice. It is not clear how to remove this restriction assuming only the abstract properties of ENDSGs. However, at the cost of one additional tight reduction to (a slight variant of) the Bilinear Decisional Diffie-Hellman (BDDH) assumption (dubbed s-BDDH), full (μ, q)-IBE-IND-CPA security can be shown.

A (subgroup) variant of the BDDH assumption (s-BDDH). For proving full security of the IBE scheme in the multi-instance, multi ciphertext setting the following assumption is required.

Definition 40. For any PPT adversary D, We say that the *subgroup Bilinear Decisional Diffie-Hellman (s-BDDH) assumption* holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$ if

$$\begin{aligned}\mathsf{Adv}^{\text{s-bddh}}_{\mathsf{ENDSG},\mathsf{Grp},D}(k,n) := |\Pr\Big[D(pp, \mathbf{g}, \mathbf{g}^a, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^a, \widehat{g}_0^b, \widehat{h}, \widehat{h}^b, \widehat{h}^c, e(\widehat{g}_0, \widehat{h})^{abc}) = 1\Big] \\ - \Pr\Big[D(pp, \mathbf{g}, \mathbf{g}^a, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^a, \widehat{g}_0^b, \widehat{h}, \widehat{h}^b, \widehat{h}^c, e(\widehat{g}_0, \widehat{h})^z) = 1\Big]|\end{aligned}$$

is negligible in k, for $(pp, sp) \leftarrow \mathsf{SampP}(1^k, n)$, for $\mathbf{g} \leftarrow \mathsf{SampG}(pp)$, for $\widehat{\mathbf{g}} = (\widehat{g}_0, \ldots, \widehat{g}_n) \leftarrow \widehat{\mathsf{SampG}}(pp, sp)$, for $\widehat{h}$ specified in sp, for e specified in pp, and for (uniform) $a, b, c, z \leftarrow \mathbb{Z}^*_N$.

In a nutshell, [HKS15] sets up $e(\widehat{g}_0^s, \widehat{h}^\gamma) = e(g,g)^{abc}$ for a generator g and random exponents a, b, c with $\widehat{g}_0^s = g^a$ and $\widehat{h}^\gamma = g^{bc}$. The s-BDDH assumption now states that $e(g,g)^{abc}$ looks random even given g, g^a, g^b, g^c. Furthermore, by the random self-reducibility of s-BDDH, the corresponding reduction is tight.

For more details, we refer to [HKS15] and [Str15]. This leads to the following corollary:

Corollary 2 (Full (μ, q)-IBE-IND-CPA security of IBE)**.** Let Grp be a group generation algorithm as defined in Definition 3. If ENDSG is an ENDSG system, s-BDDH holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$, and H is a universal hash function, then IBE is (μ, q)-IBE-IND-CPA-secure. Concretely, for any PPT adversary A with at most $q' = q'(k)$ extraction queries per instance and running time t in the (μ, q)-IBE-IND-CPA security experiment with IBE, there are distinguishers D_1 on LS1, D_2 on LS2, D_3 on NH, and D_4 on s-BDDH with running times $t'_1 \approx t'_2 \approx t'_3 \approx t'_4 \approx t + \mathbf{O}(\mu n k^c (q + q'))$, respectively, some constant $c \in \mathbb{N}$, with

$$\begin{aligned}\mathsf{Adv}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n) \leq\ & \mathsf{Adv}^{\mathrm{ls1}}_{\mathsf{ENDSG},\mathsf{Grp},D_1}(k,2n) + 2n \cdot \mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG},\mathsf{Grp},D_2}(k,2n) \\ & + n \cdot \mathsf{Adv}^{\mathrm{nh}}_{\mathsf{ENDSG},\mathsf{Grp},D_3}(k,2n,\mu q') + \mathsf{Adv}^{\text{s-bddh}}_{\mathsf{ENDSG},\mathsf{Grp},D_4}(k,2n) \\ & + \mu q \cdot \mathbf{O}(2^{-k}), \end{aligned} \tag{5.2}$$

for group generator Grp defined as above.

5.5 Instantiation of ENDSG from Composite-Order Groups

In this section we present a concrete instantiation of extended nested dual system groups from composite-order pairing-friendly groups. This is the main result in this thesis related to identity-based encryption with almost tight security in the multi-instance, multi-ciphertext setting. Especially, the relation to resulting instantiations of signature schemes in the multi-user setting with a security loss in $\mathbf{O}(k)$.

Assumptions in Groups with Composite-Order. We slightly modify two (known) dual system assumptions (see DS1, DS3 below, and [CW13]) and define one (new) dual system assumption (see DS2 below). Further, we give a dual system variant of the Bilinear Decisional Diffie-Hellman assumption, dubbed DS-BDDH, and argue that DS-BDDH implies s-BDDH from section 5.4.1.

Let $\mathsf{Grp}(1^k, 4)$ be a composite-order group generator that outputs the following group parameters

$$(\mathbb{G}, \mathbb{H} = \mathbb{G}, \mathbb{G}_T, N, e, g, g_{p_1}, g_{p_2}, g_{p_3}, g_{p_4})$$

with the composite-order groups $\mathbb{G}, \mathbb{G}_T$, each of order

$$N = p_1 \cdots p_4,$$

for pairwise-distinct k-bit primes $p_1, \ldots, p_4 \in \mathbb{P}$. Further, g_{p_i} is a generator of the subgroup $\mathbb{G}_{p_i} \subset \mathbb{G}$ of order p_i, and g is a generator of $\mathbb{G}$. More generally, we write $\mathbb{G}_q \subseteq \mathbb{G}$ for the unique subgroups of order q. We use the following assumptions in groups with composite-order:

Dual system assumption 1 (DS1). For any PPT adversary D, the function

$$\mathsf{Adv}^{\mathrm{ds1}}_{\mathsf{Grp},D}(k) := |\Pr\left[D(\mathsf{pars}, g'_{p_1}) = 1\right] - \Pr\left[D(\mathsf{pars}, g'_{p_1 p_2}) = 1\right]|$$

is negligible in k, for $(\mathbb{G}, \mathbb{G}_T, N, e, g, (g_{p_i})_{i\in[4]}) \leftarrow \mathsf{Grp}(1^k, 4)$,

$$\mathsf{pars} := (\mathbb{G}, \mathbb{G}_T, N, e, g, g_{p_1}, g_{p_3}, g_{p_4}), \text{ and } g'_{p_1} \leftarrow \mathbb{G}_{p_1},\ g'_{p_1 p_2} \leftarrow \mathbb{G}_{p_1 p_2}.$$

Dual system assumption 2 (DS2). For any PPT adversary D, the function

$$\mathsf{Adv}^{\mathrm{ds2}}_{\mathsf{Grp},D}(k) := |\Pr\left[D(\mathsf{pars}, g'_{p_1p_2}) = 1\right] - \Pr\left[D(\mathsf{pars}, g'_{p_1p_3}) = 1\right]|$$

is negligible in k, for $(\mathbb{G}, \mathbb{G}_T, N, e, g, (g_{p_i})_{i\in[4]}) \leftarrow \mathsf{Grp}(1^k, 4)$,

$$\mathsf{pars} := (\mathbb{G}, \mathbb{G}_T, N, e, g, g_{p_1}, g_{p_4}, g_{p_1p_2}, g_{p_2p_3}),$$

$$g_{p_1p_2} \leftarrow \mathbb{G}_{p_1p_2},\ g_{p_2p_3} \leftarrow \mathbb{G}_{p_2p_3},\ \text{and}\ g'_{p_1p_2} \leftarrow \mathbb{G}_{p_1p_2},\ g'_{p_1p_3} \leftarrow \mathbb{G}_{p_1p_3}.$$

Dual system assumption 3 (DS3). For any PPT adversary D, the function

$$\mathsf{Adv}^{\mathrm{ds3}}_{\mathsf{Grp},D}(k) := |\Pr\left[D(\mathsf{pars}, g_{p_2}^{xy}, g_{p_3}^{xy}) = 1\right] - \Pr\left[D(\mathsf{pars}, g_{p_2}^{xy+\gamma'}, g_{p_3}^{xy+\gamma'}) = 1\right]|$$

is negligible in k, for $(\mathbb{G}, \mathbb{G}_T, N, e, g, (g_{p_i})_{i\in[4]}) \leftarrow \mathsf{Grp}(1^k, 4)$,

$$\mathsf{pars} := (\mathbb{G}, \mathbb{G}_T, N, e, g, (g_{p_i})_i, g_{p_2}^x \widehat{X}_4, g_{p_2}^y \widehat{Y}_4, g_{p_3}^x \widetilde{X}_4, g_{p_3}^y \widetilde{Y}_4),$$

$$\widehat{X}_4, \widetilde{X}_4, \widehat{Y}_4, \widetilde{Y}_4 \leftarrow \mathbb{G}_{p_4},\ x, y, \leftarrow \mathbb{Z}_N^*,\ \text{and}\ \gamma' \leftarrow \mathbb{Z}_N^*.$$

Dual system bilinear DDH assumption (DS-BDDH). For any PPT adversary D, the function

$$\mathsf{Adv}^{\text{ds-bddh}}_{\mathsf{Grp},D}(k) := |\Pr\left[D(\mathsf{pars}, e(g_{p_2}, g_{p_2})^{abc}) = 1\right] - \Pr\left[D(\mathsf{pars}, e(g_{p_2}, g_{p_2})^{z}) = 1\right]|$$

is negligible in k, for $(\mathbb{G}, \mathbb{G}_T, N, e, g, (g_{p_i})_{i\in[4]}) \leftarrow \mathsf{Grp}(1^k, 4)$, for

$$\mathsf{pars} := (\mathbb{G}, \mathbb{G}_T, N, e, g, (g_{p_i})_i, g^a_{p_1}, g^a_{p_2}, g^b_{p_2}, g_{p_2p_4}, g^b_{p_2p_4}, g^c_{p_2p_4}),$$

$$g_{p_2p_4} \leftarrow \mathbb{G}_{p_2p_4},\ a, b, c, z \leftarrow \mathbb{Z}_N^*.$$

Lemma 13 (DS-BDDH implies s-BDDH). For any PPT adversary D with running time t on s-BDDH there is a distinguisher D' on DS-BDDH with running time $t' \approx t$ such that

$$\mathsf{Adv}^{\text{ds-bddh}}_{\mathsf{Grp},D'}(k) = \mathsf{Adv}^{\text{s-bddh}}_{\mathsf{Grp},D}(k, n),$$

for Grp as defined in Definition 3. Hence, s-BDDH holds holds relative to Grp if DS-BDDH holds relative to Grp.

Proof. **Description.** The challenge input to D' is provided as $(\mathsf{pars}, \mathbf{T})$, where $\mathbf{T}$ is either

$$\mathbf{T} = e(g_{p_2}, g_{p_2})^{abc} \ \text{ or } \ \mathbf{T} = e(g_{p_2}, g_{p_2})^{z} \leftarrow \mathbb{G}_T,$$

for

$$\mathsf{pars} = (\mathbb{G}, \mathbb{G}_T, N, e, g, (g_{p_i})_i, g^a_{p_1}, g^a_{p_2}, g^b_{p_2}, g_{p_2p_4}, g^b_{p_2p_4}, g^c_{p_2p_4}),$$

for

$$g_{p_2p_4} \xleftarrow{g} \mathbb{G}_{p_2p_4},\ \text{and for}\ a, b, c, z \leftarrow \mathbb{Z}_N^*.$$

First, D' sets the public parameters as

$$pp := (\mathbb{G}, \mathbb{H} := \mathbb{G}, \mathbb{G}_T, N, g, e, m, n, \mathsf{pars}'),$$

for $m : h' \mapsto e(g_1, h')$, $\mathsf{pars}' := (g_{p_1}, g_{p_4}, g^{\mathbf{w}}_{p_1}, h := g, h^{\mathbf{w}})$, for $\mathbf{w} \leftarrow (\mathbb{Z}_N^*)^n$, and for some integer n determined by D'.

Then, D' calls D on input

$$(pp, \mathbf{g} := (g_{p_1}^{s}, g_{p_1}^{s\cdot\mathbf{w}}), \mathbf{g}^{a}, \widehat{\mathbf{g}} := (g_{p_2}^{\hat{s}}, g_{p_2}^{\hat{s}\cdot\mathbf{w}}), \widehat{\mathbf{g}}^{a}, g_{p_2}^{b\cdot\hat{s}}, g_{p_2p_4}, g_{p_2p_4}^{b}, g_{p_2p_4}^{c}, \mathbf{T}),$$

for $s, \hat{s} \leftarrow \mathbb{Z}_N^*$.

Finally, D outputs a value which D' forwards to its own challenger.

Analysis. Note that pp is distributed as defined in s-BDDH.

If $\mathbf{T} = e(g_{p_2}, g_{p_2})^{abc}$, then

$$\Pr\left[D'(\mathsf{pars}, e(g_{p_2}, g_{p_2})^{abc}) = 1\right] = \Pr\left[D(pp, \mathbf{g}, \mathbf{g}^{a}, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^{a}, g_{p_2}^{b\hat{s}}, g_{p_2p_4}, g_{p_2p_4}^{b}, g_{p_2p_4}^{c}, e(g_{p_2}, g_{p_2})^{abc}) = 1\right]$$

follows.

Otherwise, if $\mathbf{T} = e(g_{p_2}, g_{p_2})^{z}$, then

$$\Pr\left[D'(\mathsf{pars}, e(g_{p_2}, g_{p_2})^{z}) = 1\right] = \Pr\left[D(pp, \mathbf{g}, \mathbf{g}^{a}, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^{a}, g_{p_2}^{b\hat{s}}, g_{p_2p_4}, g_{p_2p_4}^{b}, g_{p_2p_4}^{c}, e(g_{p_2}, g_{p_2})^{z}) = 1\right].$$

Hence, Lemma 13 follows. □

ENDSGs from Groups with Composite-Order. Let $\mathsf{Grp}(1^k, 4)$ be as defined in Definition 3. For simplicity, we write

$$g_i := g_{p_i} \quad \text{and} \quad g_{ij} := g_{p_i p_j},$$

for all $(i, j) \in [4] \times [4]$.

We instantiate an ENDSG $\mathsf{ENDSG}_{\mathsf{co}} = (\mathsf{SampP}, \mathsf{SampG}, \mathsf{SampH}, \widehat{\mathsf{SampG}}, \widetilde{\mathsf{SampG}})$ from composite-order groups as follows:

Parameter Sampling. $\mathsf{SampP}(1^k, n)$, on input 1^k and n, samples

$$(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, (p_i)_i, e, g, h, (g_i)_{i\in[4]}) \leftarrow \mathsf{Grp}(1^k, 4)$$

and outputs

$$pp := (\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, g, e, m, n, \mathsf{pars})$$

and

$$sp := (\widehat{h}, \widetilde{h}, \widehat{\mathsf{pars}}, \widetilde{\mathsf{pars}}),$$

for

- $m : \mathbb{H} \to \mathbb{G}_T, m : h' \mapsto e(g_1, h')$,
- $\mathsf{pars} := (g_1, g_4, g_1^{\mathbf{w}}, h, h^{\mathbf{w}} \cdot \mathbf{R_4})$, for $\mathbf{w} \leftarrow (\mathbb{Z}_N^*)^n$, $\mathbf{R}_4 \xleftarrow{g} (\mathbb{G}_{p_4})^n$,
- $\widehat{h} \xleftarrow{g} \mathbb{G}_{p_2p_4}, \widetilde{h} \xleftarrow{g} \mathbb{G}_{p_3p_4}$,
- $\widehat{pars} := (g_2, g_2^{\mathbf{w}})$, $\widetilde{pars} := (g_3, g_3^{\mathbf{w}})$.

$\mathbb{G}$-Group Sampling. $\mathsf{SampG}(pp)$, on input public parameters pp, samples a random $s \leftarrow \mathbb{Z}_N^*$ and outputs $(g_1^{s}, g_1^{s\cdot\mathbf{w}})$.

$\mathbb{H}$-Group Sampling. $\mathsf{SampH}(pp)$, on input public parameters pp, samples a random $r \leftarrow \mathbb{Z}_N^*$ and outputs $(h^{r}, h^{r\cdot\mathbf{w}} \cdot \mathbf{R}'_4)$, for $\mathbf{R}'_4 \xleftarrow{g} (\mathbb{G}_{p_4})^n$.

Semi-Functional $\mathbb{G}$-Group Sampling 1. $\widehat{\mathsf{SampG}}(pp, sp)$, on input public parameters pp and secret parameters sp, samples a random $s \leftarrow \mathbb{Z}_N^*$ and outputs $(g_2^{s}, g_2^{s\cdot\mathbf{w}})$.

Semi-Functional $\mathbb{G}$-Group Sampling 2. $\widetilde{\mathsf{SampG}}(pp, sp)$, on input public parameters pp and secret parameters sp, samples a random $s \leftarrow \mathbb{Z}_N^*$ and outputs $(g_3^{s}, g_3^{s\cdot\mathbf{w}})$.

Correctness of $\mathsf{ENDSG_{co}}$. For all $k, n \in \mathbb{N}$ and group parameters

$$(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, N, e, g, h, (g_i)_{i\in[4]}) \leftarrow \mathsf{Grp}(1^k, 4),$$

we have:

Associativity. For $\mathbf{w} = (w_1, \ldots, w_n) \in (\mathbb{Z}_N^*)^n$, for all $s, r \leftarrow \mathbb{Z}_N^*$, for $\mathbf{R'}_4 = (R'_1, \ldots, R'_n) \leftarrow (\mathbb{G}_{p_4})^n$, such that

$$(g_1^s, g_1^{s\cdot\mathbf{w}}) = \mathsf{SampG}(pp; s) \text{ and } (h^r, h^{r\cdot\mathbf{w}} \cdot \mathbf{R'}_4) = \mathsf{SampH}(pp; r),$$

it holds that

$$e(g_0, h_i) := e(g_1^s, h^{r\cdot w_i} \cdot R'_i) = e(g_1^s, h^{r\cdot w_i}) = e(g_1^{s\cdot w_i}, h^r) = e(g_i, h_0)$$

for all $i \in [n]$.

Projective. For all $s \leftarrow \mathbb{Z}_N^*$, for all $h \in \mathbb{H}$, it holds that

$$m(h)^s = e(g_1, h)^s = e(g_1^s, h) = e(g_0, h),$$

where $(g_0, \ldots, g_n) := (g_1^s, \ldots, g_1^{s\cdot w_n}) = \mathsf{SampG}(pp; s)$.

Security of $\mathsf{ENDSG_{co}}$. For all $k, n \in \mathbb{N}$, for all $(pp, sp) \leftarrow \mathsf{SampP}(1^k, n)$, we have:

Orthogonality. For $\widehat{h}, \widetilde{h}$ specified in sp, we have

$$m(\widehat{h}) = e(g_1, \widehat{h}) = e((g^{p_2p_3p_4})^{\gamma_{g_1}}, (g^{p_1p_3})^{\gamma_{\widehat{h}}}) = 1,$$
$$m(\widetilde{h}) = e(g_1, \widetilde{h}) = e((g^{p_2p_3p_4})^{\gamma_{g_1}}, (g^{p_1p_2})^{\gamma_{\widetilde{h}}}) = 1$$

for suitable exponents $\gamma_{g_1}, \gamma_{\widehat{h}}, \gamma_{\widetilde{h}} \in \mathbb{Z}_N^*$. Further, for g_1^s, $g_2^{s'}$, and $g_3^{s''}$ that are the first outputs of $\mathsf{SampG}(pp; s)$, $\widehat{\mathsf{SampG}}(pp, sp; s')$, and $\widetilde{\mathsf{SampG}}(pp, sp; s'')$, for $s, s', s'' \leftarrow \mathbb{Z}_N^*$, we have

$$e(g_1^s, \widehat{h}) = e(g_0, \widehat{h}) = 1,$$
$$e(g_1^s, \widetilde{h}) = e(g_0, \widetilde{h}) = 1,$$
$$e(g_2^{s'}, \widetilde{h}) = e(\widehat{g_0}, \widetilde{h}) = 1,$$
$$e(g_3^{s''}, \widehat{h}) = e(\widetilde{g_0}, \widehat{h}) = 1.$$

$\mathbb{G}$- and $\mathbb{H}$-Subgroups. Since g_1, g_2, and g_3 are generators of subgroups $\mathbb{G}_{p_1}$, $\mathbb{G}_{p_2}$, and $\mathbb{G}_{p_3}$ of coprime order, the outputs of SampG, $\widehat{\mathsf{SampG}}$, and $\widetilde{\mathsf{SampG}}$ are uniform over the generators, which generates nontrivial subgroups of $\mathbb{G}$ of coprime order. Since h is a generator of $\mathbb{H}$ and $\mathbf{R}'_4$ is uniform over the generators of $(\mathbb{G}_{p_4})^n$, the output of SampH is uniformly distributed over the generators of $\mathbb{H}$.

Non-Degeneracy. For the first output g_2^s of $\widehat{\mathsf{SampG}}(pp, sp; s)$ (with uniform $s \in \mathbb{Z}_N^*$), and for $\widehat{h} \in \mathbb{G}_{p_2p_3}$ as specified in sp, it holds that

$$e(g_2^s, \widehat{h}) = e(g_2, \widehat{h})^s$$

is uniformly distributed over the generators of the subgroup generated by $e(g_2, \widehat{h})$. This subgroup is nontrivial except with probability $2^{-\mathbf{\Omega}(k)}$ (over pp). Similarly, for the first output g_3^s of $\widetilde{\mathsf{SampG}}(pp, sp; s)$, it holds that

$$e(g_3^s, \widetilde{h}) = e(g_3, \widetilde{h})^s$$

is distributed uniformly over the generators of the subgroup generated by $e(g_3, \widetilde{h})$ (for uniform $s \leftarrow \mathbb{Z}_N^*$).

Left-Subgroup Indistinguishability 1. We prove the following lemma.

Lemma 14 (DS1 implies LS1). For any PPT adversary D with running time t on LS1 of $\mathsf{ENDSG_{co}}$ as defined above there is a distinguisher D' on DS1 with running time $t' \approx t$ such that

$$\mathsf{Adv}^{\mathrm{ds1}}_{\mathsf{Grp},D'}(k) = \mathsf{Adv}^{\mathrm{ls1}}_{\mathsf{ENDSG_{co}},\mathsf{Grp},D}(k,n),$$

for Grp as defined in Definition 3. Hence, LS1 holds relative to Grp if DS1 holds relative to Grp.

Proof. **Description.** The challenge input to D' is provided as $(\mathsf{pars}, \mathbf{T})$, where $\mathbf{T}$ is either

$$\mathbf{T} = g_1' \leftarrow \mathbb{G}_{p_1} \quad \text{or} \quad \mathbf{T} = g_{12}' \leftarrow \mathbb{G}_{p_1 p_2},$$

for

$$\mathsf{pars} = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_1, g_3, g_4).$$

First, D' sets the public parameters as

$$pp := (\mathbb{G}, \mathbb{H} := \mathbb{G}, \mathbb{G}_T, N, g, e, m, n, \mathsf{pars}'),$$

for

$$m : \mathbb{H} \to \mathbb{G}_T, m : h' \mapsto e(g_1, h'),$$

$$\mathsf{pars}' := (g_1, g_4, g_1^{\mathbf{w}}, h := g, h^{\mathbf{w}}), \text{ for } \mathbf{w} \leftarrow (\mathbb{Z}_N^*)^n,$$

and for some integer n determined by D'. Then, D' calls D on input $(pp, \mathbf{T}, \mathbf{T}^{\mathbf{w}})$. Finally, D outputs a value which D' forwards to its own challenger.

Analysis. Note that pp is distributed as defined in LS1.

If $\mathbf{T} = g_1'$, then

$$(g_1', (g_1')^{\mathbf{w}})$$

is distributed as the output of $\mathsf{SampG}(pp)$ as needed and, hence,

$$\Pr\left[D'(\mathsf{pars}, g_1') = 1\right] = \Pr\left[D(pp, (g_1', (g_1')^{\mathbf{w}})) = 1\right]$$

follows.

Otherwise, if $\mathbf{T} = g_{12}'$, then

$$(g_{12}', (g_{12}')^{\mathbf{w}})$$

is distributed as $\mathsf{SampG}(pp) \cdot \widehat{\mathsf{SampG}}(pp, sp)$, for suitable sp, as desired and, hence, we have that

$$\Pr\left[D'(\mathsf{pars}, g_{12}') = 1\right] = \Pr\left[D(pp, (g_{12}', (g_{12}')^{\mathbf{w}})) = 1\right].$$

As a consequence, Lemma 14 follows. □

Left-Subgroup Indistinguishability 2. We prove the following lemma.

Lemma 15 (DS2 implies LS2). For any PPT adversary D with running time t on LS2 of $\mathsf{ENDSG_{co}}$ defined as above there is a distinguisher D' on DS2 with running time $t' \approx t$ such that

$$\mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG_{co}},\mathsf{Grp},D}(k,n) = \mathsf{Adv}^{\mathrm{ds2}}_{\mathsf{Grp},D'}(k),$$

for Grp as defined in Definition 3. Hence, LS2 holds relative to Grp if DS2 holds relative to Grp.

Proof. **Description.** The challenge input to D' is provided as $(\mathsf{pars}, \mathbf{T})$, where $\mathbf{T}$ is either

$$\mathbf{T} = g'_{12} \leftarrow \mathbb{G}_{p_1 p_2}$$

or

$$\mathbf{T} = g'_{13} \leftarrow \mathbb{G}_{p_1 p_3},$$

for

$$\mathsf{pars} = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_1, g_4, g_{12}, g_{23}).$$

First, D' defines the public parameters as

$$pp := (\mathbb{G}, \mathbb{H} := \mathbb{G}, \mathbb{G}_T, N, g, e, m, n, \mathsf{pars}'),$$

for

$$m : \mathbb{H} \to \mathbb{G}_T, m : h' \mapsto e(g_1, h'), \text{ and}$$
$$\mathsf{pars}' := (g_1, g_4, g_1^{\mathbf{w}}, h := g, h^{\mathbf{w}}), \text{ for } \mathbf{w} \leftarrow (\mathbb{Z}_N^*)^n,$$

and for some integer n determined by D'.

Then, D' calls D on input

$$(pp, g_{23} g_4^{\gamma}, g_{12}, \mathbf{T}, \mathbf{T}^{\mathbf{w}})$$

for $\gamma \leftarrow \mathbb{Z}_N^*$. Finally, D outputs a value which is forwarded by D' to its own challenger.

Analysis. Note that pp is distributed as defined in LS2.

If $\mathbf{T} = g'_{12}$, then

$$(g'_{12}, (g'_{12})^{\mathbf{w}})$$

is distributed as $\mathsf{SampG}(pp) \cdot \widehat{\mathsf{SampG}}(pp, sp)$, for suitable sp, as needed and, hence, we have that

$$\Pr\left[D'(\mathsf{pars}, g'_{12}) = 1\right] = \Pr\left[D(pp, g_{23} g_4^{\gamma}, g_{12}, (g'_{12}, (g'_{12})^{\mathbf{w}})) = 1\right]$$

follows.

Otherwise, if $\mathbf{T} = g'_{13}$, then

$$(g'_{13}, (g'_{13})^{\mathbf{w}})$$

is distributed as $\mathsf{SampG}(pp) \cdot \widetilde{\mathsf{SampG}}(pp, sp)$, for suitable sp, as desired and, hence,

$$\Pr\left[D'(\mathsf{pars}, g'_{13}) = 1\right] = \Pr\left[D(pp, g_{23} g_4^{\gamma}, g_{12}, (g'_{13}, (g'_{13})^{\mathbf{w}})) = 1\right]$$

holds. As a consequence, Lemma 15 follows. □

Nested-Hiding Indistinguishability. We prove the following lemma.

Lemma 16 (DS3 implies NH). For any PPT adversary D with running time t on NH of $\mathsf{ENDSG}_{\mathsf{co}}$ there is a distinguisher D' on DS3 with running time $t' \approx t$ such that

$$\mathsf{Adv}^{\mathrm{nh}}_{\mathsf{ENDSG}_{\mathsf{co}}, \mathsf{Grp}, D}(k, n, q') \leq \mathsf{Adv}^{\mathrm{ds3}}_{\mathsf{Grp}, D'}(k),$$

for all $q' \in \mathbb{N}$. Hence, NH holds relative to Grp if DS3 holds relative to Grp.

Proof. The proof follows the same strategy as shown in Chen and Wee's work [CW13] except that we have to integrate two coprime-order semi-functional generators $\widehat{h}$ and $\widetilde{h}$ instead of just one as in [CW13].

Description. The challenge input to D' is provided as $(\mathsf{pars}, \mathbf{T})$, where $\mathbf{T} := (\widehat{\mathbf{T}}, \widetilde{\mathbf{T}})$ is either

$$\mathbf{T} = (\widehat{\mathbf{T}}, \widetilde{\mathbf{T}}) = (g_2^{xy}, g_3^{xy})$$

or

$$\mathbf{T} = (\widehat{\mathbf{T}}, \widetilde{\mathbf{T}}) = (g_2^{xy+\gamma'}, g_3^{xy+\gamma'}),$$

for

$$\mathsf{pars} =: (\mathbb{G}, \mathbb{G}_T, N, e, g_1, g_2, g_3, g_4, g_2^x \widehat{X}_4, g_2^y \widehat{Y}_4, g_3^x \widetilde{X}_4, g_3^y \widetilde{Y}_4),$$

for $\widehat{X}_4, \widehat{Y}_4, \widetilde{X}_4, \widetilde{Y}_4 \xleftarrow{g} \mathbb{G}_{p_4}$, $x, y \leftarrow \mathbb{Z}_N^*$, and for $\gamma' \leftarrow \mathbb{Z}_N^*$.

Furthermore, D' receives an auxiliary input $i \in [\lfloor \frac{n}{2} \rfloor]$, for some integer $n \in \mathbb{N}$ determined by D'. First, D' samples $r, \hat{r}, \tilde{r}, \hat{s}, \tilde{s} \leftarrow \mathbb{Z}_N^*$, $\mathbf{R'}_4 \xleftarrow{g} (\mathbb{G}_{p_4})^n$, $\mathbf{w'} \leftarrow (\mathbb{Z}_N^*)^n$, and sets

$$h := (g_1 g_2 g_3 g_4)^r, \qquad \widehat{h} := (g_2 g_4)^{\hat{r}}, \qquad \widetilde{h} := (g_3 g_4)^{\tilde{r}},$$
$$\widehat{\mathbf{g}}_{-(2i-1)} := (g_2^{\hat{s}}, g_2^{\hat{s}\mathbf{w'}})_{-(2i-1)}, \qquad \widetilde{\mathbf{g}}_{-2i} := (g_3^{\tilde{s}}, g_3^{\tilde{s}\mathbf{w'}})_{-(2i)},$$

where h, $\widehat{h}$, and $\widetilde{h}$ are generators of $\mathbb{G}$, $\mathbb{G}_{p_2 p_4}$, and $\mathbb{G}_{p_3 p_4}$, respectively.

Then, D' defines public parameters as

$$pp := (\mathbb{G}, \mathbb{H} := \mathbb{G}, \mathbb{G}_T, N, g, e, n, m, \mathsf{pars}'),$$

for

$$m : \mathbb{H} \to \mathbb{G}_T, m : h' \mapsto e(g_1, h')$$

and

$$\begin{aligned} \mathsf{pars}' &:= (g_1, g_4, g_1^{\mathbf{w}'}, h, h^{\mathbf{w}'}(g_2^y \widehat{Y}_4)^{r\mathbf{e}_{2i-1}}(g_3^y \widetilde{Y}_4)^{r\mathbf{e}_{2i}} \mathbf{R'}_4) \\ &= (g_1, g_4, g_1^{\mathbf{w}}, h, h^{\mathbf{w}} \mathbf{R}_4), \end{aligned}$$

where $\mathbf{e}_j$ is the j-th unit vector of length n. Hence, implicitly, we have

$$\mathbf{w} = \begin{cases} \mathbf{w}' \mod p_1 p_4 \\ \mathbf{w}' + y \cdot \mathbf{e}_{2i-1} \mod p_2 \\ \mathbf{w}' + y \cdot \mathbf{e}_{2i} \mod p_3 \end{cases} \quad \text{and} \quad \mathbf{R}_4 = \mathbf{R'}_4 + \widehat{Y}_4^r \cdot \mathbf{e}_{2i-1} + \widetilde{Y}_4^r \cdot \mathbf{e}_{2i}.$$

Now, by running the algorithm from [CW14, Lemma 6] on input

$$(1^{q'}, (g_2, g_4, g_2^x \widehat{X}_4, g_2^y \widehat{Y}_4, \widehat{\mathbf{T}}))$$

and on input

$$(1^{q'}, (g_3, g_4, g_3^x \widetilde{X}_4, g_3^y \widetilde{Y}_4, \widetilde{\mathbf{T}})),$$

D' generates tuples

$$(g_2^{\hat{r}_j} \widehat{X}_{4,j}, \widehat{\mathbf{T}}_j)_{j=1}^{q'} \text{ and } (g_3^{\tilde{r}_j} \widetilde{X}_{4,j}, \widetilde{\mathbf{T}}_j)_{j=1}^{q'},$$

respectively, where

$$\widehat{\mathbf{T}}_j = \begin{cases} g_2^{\hat{r}_j y} \cdot \widehat{Y}_{4,j}, & \text{if } \widehat{\mathbf{T}} = g_2^{xy} \\ g_2^{\hat{r}_j y} \cdot \widehat{Y}_{4,j} \cdot g_2^{\hat{\gamma'}_j}, & \text{if } \widehat{\mathbf{T}} = g_2^{xy+\gamma'} \end{cases}$$

and

$$\widetilde{\mathbf{T}}_j = \begin{cases} g_3^{\tilde{r}_j y} \cdot \widetilde{Y}_{4,j}, & \text{if } \widetilde{\mathbf{T}} = g_3^{xy} \\ g_3^{\tilde{r}_j y} \cdot \widetilde{Y}_{4,j} \cdot g_3^{\tilde{\gamma'}_j}, & \text{if } \widetilde{\mathbf{T}} = g_3^{xy+\gamma'}. \end{cases}$$

Further, D' samples $r'_j \leftarrow \mathbb{Z}_N^*$, $\mathbf{X'}_{4,j} \xleftarrow{g} (\mathbb{G}_{p_4})^n$, for all $j \in [q']$, and calls D on input

$$(pp, \widehat{h}, \widetilde{h}, \widehat{\mathbf{g}}_{2i-1}, \widetilde{\mathbf{g}}_{2i}, (\mathbf{T}_1, \ldots, \mathbf{T}_{q'})),$$

where

$$\begin{aligned}
\mathbf{T}_j &= (h^{r'_j} \cdot g_2^{\hat{r}_j} \widehat{X}_{4,j} \cdot g_3^{\tilde{r}_j} \widetilde{X}_{4,j}, (h^{r'_j} \cdot g_2^{\hat{r}_j} \widehat{X}_{4,j} \cdot g_3^{\tilde{r}_j} \widetilde{X}_{4,j})^{\mathbf{w}'} \cdot \\
&\quad ((g_2^y \widehat{Y}_4)^{r'_j r} \widehat{\mathbf{T}}_j)^{\mathbf{e}_{2i-1}} \cdot ((g_3^y \widetilde{Y}_4)^{r'_j r} \widetilde{\mathbf{T}}_j)^{\mathbf{e}_{2i}} \mathbf{X'}_{4,j}) \\
&= \begin{cases} (h^{r_j}, h^{r_j \cdot \mathbf{w}} \cdot \mathbf{X}_{4,j}) & \text{if } \widehat{\mathbf{T}}_j = g_2^{\hat{r}_j y} \cdot \widehat{Y}_{4,j} \\ & \text{and } \widetilde{\mathbf{T}}_j = g_3^{\tilde{r}_j y} \cdot \widetilde{Y}_{4,j} \\ (h^{r_j}, h^{r_j \cdot \mathbf{w}} \cdot g_2^{\hat{\gamma}_j \mathbf{e}_{2i-1}} \cdot g_3^{\tilde{\gamma}_j \mathbf{e}_{2i}} \cdot \mathbf{X}_{4,j}) & \text{if } \widehat{\mathbf{T}}_j = g_2^{\hat{r}_j y} \cdot \widehat{Y}_{4,j} \cdot g_2^{\hat{\gamma}_j} \\ & \text{and } \widetilde{\mathbf{T}}_j = g_3^{\tilde{r}_j y} \cdot \widetilde{Y}_{4,j} \cdot g_3^{\tilde{\gamma}_j} \end{cases}
\end{aligned}$$

for $h^{r_j} := h^{r'_j} \cdot g_2^{\hat{r}_j} \widehat{X}_{4,j} \cdot g_3^{\tilde{r}_j} \widetilde{X}_{4,j}$ and $\mathbf{X}_{4,j} := \mathbf{X'}_{4,j} + \widehat{Y}_4^{r'_j r} \mathbf{e}_{2i-1} + \widetilde{Y}_4^{r'_j r} \mathbf{e}_{2i}$ implicitly and $\mathbf{w}$ as above.

Analysis. Note that pp is distributed as defined in NH.
If $\mathbf{T} = (g_2^{xy}, g_3^{xy})$, then

$$\widehat{\mathbf{T}}_j = g_2^{\hat{r}_j y} \cdot \widehat{Y}_{4,j}$$

and

$$\widetilde{\mathbf{T}}_j = g_3^{\tilde{r}_j y} \cdot \widetilde{Y}_{4,j},$$

for all $j \in [q']$, and, thus, $(\mathbf{T}_1, \ldots, \mathbf{T}_{q'})$ is distributed as $(\mathbf{h}_1, \ldots, \mathbf{h}_{q'})$, for suitable sp, as needed.
Otherwise, if $\mathbf{T} = (g_2^{xy+\gamma'}, g_3^{xy+\gamma'})$, then

$$\widehat{\mathbf{T}}_j = g_2^{\hat{r}_j y} \cdot \widehat{Y}_{4,j} \cdot g_2^{\hat{\gamma}_j}$$

and

$$\widetilde{\mathbf{T}}_j = g_3^{\tilde{r}_j y} \cdot \widetilde{Y}_{4,j} \cdot g_3^{\tilde{\gamma}_j},$$

for all $j \in [q']$, and, thus, $(\mathbf{T}_1, \ldots, \mathbf{T}_{q'})$ is distributed as $(\mathbf{h'}_1, \ldots, \mathbf{h'}_{q'})$, for suitable sp, since $(\widehat{h}, g_2^{\hat{\gamma}_j} \cdot \widehat{Y}_{4,j})$ and $(\widetilde{h}, g_3^{\tilde{\gamma}_j} \cdot \widetilde{Y}_{4,j})$ are identically distributed as $(\widehat{h}, (\widehat{h})^{\hat{\gamma}_j} \cdot \widehat{Y}_{4,j})$ and $(\widetilde{h}, (\widetilde{h})^{\tilde{\gamma}_j} \cdot \widetilde{Y}_{4,j})$, respectively, for $\hat{\gamma}_j, \tilde{\gamma}_j \leftarrow \mathbb{Z}_N^*, \widehat{Y}_{4,j}, \widetilde{Y}_{4,j} \xleftarrow{g} \mathbb{G}_{p_4}$, for all $j \in [q']$. □

Result. Now, we have shown that our concrete instantiation $\mathsf{ENDSG_{co}}$ from composite-order groups satisfies all required properties of extended nested dual system groups as defined in section 5.4. If the dual system assumptions DS1, DS2 and DS3 introduced in this section holds relative to a group generation algorithm $\mathsf{Grp}(\cdot)$ defined in Definition 3 (i.e., in composite-order pairing-friendly groups) then Lemma 14, Lemma 15 and Lemma 16 show the implications for LS1, LS2 and NH. Together with a universal hash function this allows to apply Theorem 12 to achieve a weakly (μ, q)-IBE-IND-CPA-secure IBE scheme. Since Lemma 13 show that our dual system variant of the BDDH assumption implies the s-BDDH assumption from section 5.4.1, we are able to apply Corollary 2 to achieve full (μ, q)-IBE-IND-CPA security.

5.6 State-of-the-Art

Our construction is built in composite-order groups and one open problem stated in [HKS15] was, if it is possible to give a realization of extended nested dual system groups in prime order groups, which are more efficient. In groups with composite-order it must be ensured, that it is not possible to factorize efficiently. For this reason, the parameters must be larger.

5.6.1 Almost Tight IBE Security in Prime-Order Groups

Shortly after our result for composite-order groups, [AHY15] and [GCDCT16], independently, gave prime-order solutions in the multi-instance, multi-ciphertext setting. In [GCDCT16] they follow the extended nested dual system groups strategy and proposed two constructions. The first one is under a standard assumption. The second one is an improved version of the first one (in terms of efficiency regarding ciphertext/key size and encryption/decryption cost) but relies on a non-standard assumption.

In [AHY15] their approach was different. They introduced a new notion, *broadcast encoding*, which is compatible with composite-order and prime-order groups. Thus, they provided a generic framework to build almost tightly secure ordinary IBE schemes and IBE schemes with additional features. Their scheme achieves a similar efficiency as in [GCDCT16] but also under a non-standard assumption. This suggested that there is a trade-off between efficiency and the strength of a complexity assumption. The question arises if it is possible to construct an (almost) tightly secure IBE scheme in the multi-instance, multi-ciphertext setting under a standard assumption with similar efficiency as in [GCDCT16] and [AHY15].

5.6.2 Recent Improvements

In [GDCC16] this question was answered in the affirmative. Their scheme offers the same efficiency as the second one in [GCDCT16], but relies on a standard assumption. They revisted the almost tigthly secure IBE scheme of Blazy et al. [BKP14] and combine it with the framework of extended nested dual system groups and the ideas in [GCDCT16] to extend it to the multi-instance, multi-ciphertext setting. Subsequently, [LYWY17] improves this result for Blazy et al.'s scheme.

Another important improvement regarding the tightness of the security reduction in the multi-instance, multi-ciphertext setting was given in [CGW17]. Their scheme achieves constant size master secret keys, ciphertexts and user secret keys and has a fast decryption algorithm. Their security reduction loss is linear in $\log(q)$, i.e., in $\mathbf{O}(\log(q))$, where q is the upper bound of user secret keys and ciphertexts queries per instance. For concrete values, e.g. $q = 2^{30}$ and $n = 128$ this is a tighter reduction than in previous works. However, the security relies on a non-standard assumption in composite-order groups.

Regarding chosen-ciphertext security [HJP18] recently constructed the first IBE scheme, which is almost tightly secure against chosen ciphertext attacks in the multi-instance, multi-ciphertext setting.

6 Concluding Remarks

Digital signature schemes, and all its variants and related cryptographic concepts, such as aggregate signature schemes and identity-based encryption schemes, are a big and important branch in cryptographic research.

The design and construction of efficient schemes, in terms of short key and signature size and in terms of fast signing and verification computations, is essential for the use in practice. If this can be combined with strong security requirements proven in a realistic model under reliable and simple assumptions, this is a great achievement. According to the latest state-of-the-art, the multi-user scenario is a more accurate representation of real-world applications.

Furthermore, since the quality of a security reduction has an impact on the recommended key lengths in practice, a tight reduction leads to shorter keys and parameters.

If you take all this together an efficient digital signature scheme proven strongly and tightly secure under standard assumptions in the multi-user model is preferred. Depending on the type of signature schemes (e.g., structure-preserving signatures) this often requires a trade-off and research is still far away to achieve all of this.

We hope one step towards this goal was made in this thesis. We constructed a very efficient digital signature scheme proven strongly secure under a standard assumption in the standard model. We also realized an almost tight security reduction in a multi-user, multi-challenge setting for identity-based encryption, closely related to digital signatures.

Bibliography

[ADKNO13] Masayuki Abe, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki and Miyako Ohkubo. 'Tagged One-Time Signatures: Tight Security and Optimal Tag Size'. In: *Public-Key Cryptography – PKC 2013*. Ed. by Kaoru Kurosawa and Goichiro Hanaoka. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 312–331.

[AGH10] Jae Hyun Ahn, Matthew Green and Susan Hohenberger. 'Synchronized aggregate signatures: new definitions, constructions and applications'. In: *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*. 2010, pp. 473–484. DOI: 10.1145/1866307.1866360. URL: https://doi.org/10.1145/1866307.1866360.

[AHY15] Nuttapong Attrapadung, Goichiro Hanaoka and Shota Yamada. 'A Framework for Identity-Based Encryption with Almost Tight Security'. In: *Advances in Cryptology – ASIACRYPT 2015*. Ed. by Tetsu Iwata and Jung Hee Cheon. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 521–549.

[AS15] Jacob Alperin-Sheriff. 'Short Signatures with Short Public Keys from Homomorphic Trapdoor Functions'. In: *Public-Key Cryptography – PKC 2015*. Ed. by Jonathan Katz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 236–255.

[ASA17] Jacob Alperin-Sheriff and Daniel Apon. 'Weak is Better: Tightly Secure Short Signatures from Weak PRFs'. In: *IACR ePrint* 2017.IACR ePrint (2017).

[Bad14] Christoph Bader. 'Efficient Signatures with Tight Real World Security in the Random-Oracle Model'. In: *Cryptology and Network Security*. Ed. by Dimitris Gritzalis, Aggelos Kiayias and Ioannis Askoxylakis. Cham: Springer International Publishing, 2014, pp. 370–383.

[BB04a] Dan Boneh and Xavier Boyen. 'Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles'. In: *Advances in Cryptology - EUROCRYPT 2004*. Ed. by Christian Cachin and Jan L. Camenisch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 223–238.

[BB04b] Dan Boneh and Xavier Boyen. 'Secure Identity Based Encryption Without Random Oracles'. In: *Advances in Cryptology – CRYPTO 2004*. Ed. by Matt Franklin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 443–459.

[BB04c] Dan Boneh and Xavier Boyen. 'Short Signatures Without Random Oracles'. In: *Advances in Cryptology - EUROCRYPT 2004*. Ed. by Christian Cachin and Jan L. Camenisch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 56–73.

[BB08] Dan Boneh and Xavier Boyen. 'Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups'. In: *Journal of Cryptology* 21.2 (2008), pp. 149–177.

[BBM00] Mihir Bellare, Alexandra Boldyreva and Silvio Micali. 'Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements'. In: *Advances in Cryptology — EUROCRYPT 2000*. Ed. by Bart Preneel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 259–274.

[BDJR97] Mihir Bellare, Anand Desai, Eron Jokipii and Phillip Rogaway. 'A Concrete Security Treatment of Symmetric Encryption'. In: *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*. IEEE. 1997, pp. 394–403.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval and Phillip Rogaway. 'Relations Among Notions of Security for Public-Key Encryption Schemes'. In: *Advances in Cryptology — CRYPTO '98*. Ed. by Hugo Krawczyk. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 26–45.

[BF01] Dan Boneh and Matt Franklin. 'Identity-Based Encryption from the Weil Pairing'. In: *Advances in Cryptology — CRYPTO 2001*. Ed. by Joe Kilian. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 213–229.

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn and Hovav Shacham. 'Aggregate and Verifiably Encrypted Signatures from Bilinear Maps'. In: *EUROCRYPT 2003*. Ed. by Eli Biham. Vol. 2656. LNCS. Springer, May 2003, pp. 416–432. DOI: 10.1007/3-540-39200-9_26.

[BHJKL15] Christoph Bader, Dennis Hofheinz, Tibor Jager, Eike Kiltz and Yong Li. 'Tightly-Secure Authenticated Key Exchange'. In: *Theory of Cryptography*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 629–658.

[BHJKS13] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch and Christoph Striecks. *Confined Guessing: New Signatures From Standard Assumptions*. Cryptology ePrint Archive, Report 2013/171. http://eprint.iacr.org/2013/171. 2013.

[BHJKSS13] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, Jae Hong Seo and Christoph Striecks. 'Practical Signatures from Standard Assumptions'. In: *EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, May 2013, pp. 461–485. DOI: 10.1007/978-3-642-38348-9_28.

[BIM18] Thais Bardini Idalino and Lucia Moura. 'Efficient Unbounded Fault-Tolerant Aggregate Signatures Using Nested Cover-Free Families'. In: *Combinatorial Algorithms*. Ed. by Costas Iliopoulos, Hon Wai Leong and Wing-Kin Sung. Cham: Springer International Publishing, 2018, pp. 52–64.

[BKP14] Olivier Blazy, Eike Kiltz and Jiaxin Pan. '(Hierarchical) Identity-Based Encryption from Affine Message Authentication'. In: *Advances in Cryptology – CRYPTO 2014*. Ed. by Juan A. Garay and Rosario Gennaro. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 408–425.

[BL16] Xavier Boyen and Qinyi Li. 'Towards Tightly Secure Short Signature and IBE'. In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 498.

[BLS04] Dan Boneh, Ben Lynn and Hovav Shacham. 'Short Signatures from the Weil Pairing'. In: vol. 17. 4. 2004, pp. 297–319.

[BNN07] Mihir Bellare, Chanathip Namprempre and Gregory Neven. 'Unrestricted Aggregate Signatures'. In: *Automata, Languages and Programming*. Ed. by Lars Arge, Christian Cachin, Tomasz Jurdziński and Andrzej Tarlecki. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 411–422.

[Bol09] Alexandra Boldyreva. 'Strengthening Security of RSA-OAEP'. In: *Topics in Cryptology – CT-RSA 2009*. Ed. by Marc Fischlin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 399–413.

[Boy10] Xavier Boyen. 'Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More'. In: *Public Key Cryptography – PKC 2010*. Ed. by Phong Q. Nguyen and David Pointcheval. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 499–517.

[BR93] Mihir Bellare and Phillip Rogaway. 'Random Oracles are Practical: A Paradigm for Designing Efficient Protocols'. In: *Proceedings of the 1st ACM conference on Computer and communications security*. ACM. 1993, pp. 62–73.

[CCMT09] Claude Castelluccia, Aldar CF Chan, Einar Mykletun and Gene Tsudik. 'Efficient and Provably Secure Aggregation of Encrypted Data in Wireless Sensor Networks'. In: *ACM Transactions on Sensor Networks (TOSN)* 5.3 (2009), p. 20.

[CD96] Ronald Cramer and Ivan Damgård. 'New Generation of Secure and Practical RSA-Based Signatures'. In: *Advances in Cryptology — CRYPTO '96*. Ed. by Neal Koblitz. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 173–185.

[CFHIZ07] Yang Cui, Eiichiro Fujisaki, Goichiro Hanaoka, Hideki Imai and Rui Zhang. 'Formal Security Treatments for Signatures from Identity-Based Encryption'. In: (2007). Ed. by Willy Susilo, Joseph K. Liu and Yi Mu, pp. 218–227.

[CGH98] Ran Canetti, Oded Goldreich and Shai Halevi. 'The Random Oracle Methodology, Revisited (Preliminary Version)'. In: *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*. 1998, pp. 209–218. DOI: 10.1145/276698.276741. URL: https://doi.org/10.1145/276698.276741.

[CGW17] Jie Chen, Junqing Gong and Jian Weng. 'Tightly Secure IBE Under Constant-Size Master Public Key'. In: *Public-Key Cryptography – PKC 2017*. Ed. by Serge Fehr. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 207–231.

[CHK07] Ran Canetti, Shai Halevi and Jonathan Katz. 'A Forward-Secure Public-Key Encryption Scheme'. In: vol. 20. 3. 2007, pp. 265–294.

[CKS08] David Cash, Eike Kiltz and Victor Shoup. 'The Twin Diffie-Hellman Problem and Applications'. In: *Advances in Cryptology – EUROCRYPT 2008*. Ed. by Nigel Smart. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 127–145.

[CL04] Jan Camenisch and Anna Lysyanskaya. 'Signature Schemes and Anonymous Credentials from Bilinear Maps'. In: *Advances in Cryptology – CRYPTO 2004*. Ed. by Matt Franklin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 56–72.

[Coc01] Clifford Cocks. 'An Identity Based Encryption Scheme Based on Quadratic Residues'. In: *Cryptography and Coding*. Ed. by Bahram Honary. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 360–363.

[Cor00] Jean-Sébastien Coron. 'On the Exact Security of Full Domain Hash'. In: *Advances in Cryptology — CRYPTO 2000*. Ed. by Mihir Bellare. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 229–235.

[CS00] Ronald Cramer and Victor Shoup. 'Signature Schemes Based on the Strong RSA Assumption'. In: *ACM Transactions on Information and System Security (TISSEC)* 3.3 (2000), pp. 161–185.

[CW13] Jie Chen and Hoeteck Wee. 'Fully, (Almost) Tightly Secure IBE and Dual System Groups'. In: *Advances in Cryptology – CRYPTO 2013*. Ed. by Ran Canetti and Juan A. Garay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 435–460.

[CW14] Jie Chen and Hoeteck Wee. 'Dual System Groups and its Applications-Compact HIBE and More.' In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 265.

[DH76] Whitfield Diffie and Martin Hellman. 'New Directions in Cryptography'. In: *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654.

[DLVY01] AG D'yachkov, VS Lebedev, PA Vilenkin and SM Yekhanin. 'Cover-Free Families and Superimposed Codes: Constructions, Bounds and Applications to Cryptography and Group Testing'. In: *Information Theory, 2001. Proceedings. 2001 IEEE International Symposium on*. IEEE. 2001, p. 117.

[DM14] Léo Ducas and Daniele Micciancio. 'Improved Short Lattice Signatures in the Standard Model'. In: *Advances in Cryptology – CRYPTO 2014*. Ed. by Juan A. Garay and Rosario Gennaro. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 335–352.

[DMR00] Arkadii G. D'yachkov, Anthony J. Macula and Vyacheslav V. Rykov. 'New Applications and Results of Superimposed Code Theory Arising from the Potentialities of Molecular Biology'. In: *Numbers, Information and Complexity*. Ed. by Ingo Althöfer, Ning Cai, Gunter Dueck, Levon Khachatrian, Mark S. Pinsker, Andras Sárközy, Ingo Wegener and Zhen Zhang. Boston, MA: Springer US, 2000, pp. 265–282.

[DVPS14] A. G. D'yachkov, I. V. Vorob'ev, N. A. Polyansky and V. Yu. Shchukin. 'Bounds on the Rate of Disjunctive Codes'. In: *Problems of Information Transmission* 50.1 (2014), pp. 27–56.

[Fis02] Marc Fischlin. 'The Cramer-Shoup Strong-RSA Signature Scheme Revisited'. In: *Public Key Cryptography — PKC 2003*. Ed. by Yvo G. Desmedt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 116–129.

[FJS14] Nils Fleischhacker, Tibor Jager and Dominique Schröder. 'On Tight Security Proofs for Schnorr Signatures'. In: *Advances in Cryptology – ASIACRYPT 2014*. Ed. by Palash Sarkar and Tetsu Iwata. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 512–531.

[FLS12] Marc Fischlin, Anja Lehmann and Dominique Schröder. 'History-Free Sequential Aggregate Signatures'. In: *Security and Cryptography for Networks*. Ed. by Ivan Visconti and Roberto De Prisco. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 113–130.

[Für96] Zoltán Füredi. 'On r-Cover-Free Families'. In: *Journal of Combinatorial Theory, Series A* 73.1 (1996), pp. 172–173.

[GCDCT16] Junqing Gong, Jie Chen, Xiaolei Dong, Zhenfu Cao and Shaohua Tang. 'Extended Nested Dual System Groups, Revisited'. In: *Public-Key Cryptography – PKC 2016*. Ed. by Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano and Bo-Yin Yang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 133–163.

[GDCC16] Junqing Gong, Xiaolei Dong, Jie Chen and Zhenfu Cao. 'Efficient IBE with Tight Reduction to Standard Assumption in the Multi-challenge Setting'. In: *Advances in Cryptology – ASIACRYPT 2016*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 624–654.

[Gen06] Craig Gentry. 'Practical Identity-Based Encryption Without Random Oracles'. In: *Advances in Cryptology - EUROCRYPT 2006*. Ed. by Serge Vaudenay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 445–464.

[GH09] Craig Gentry and Shai Halevi. 'Hierarchical Identity Based Encryption with Polynomially Many Levels'. In: *Theory of Cryptography*. Ed. by Omer Reingold. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 437–456.

[GHR99] Rosario Gennaro, Shai Halevi and Tal Rabin. 'Secure Hash-and-Sign Signatures Without the Random Oracle'. In: *Advances in Cryptology — EUROCRYPT '99*. Ed. by Jacques Stern. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 123–139.

[GJ18] Kristian Gjøsteen and Tibor Jager. 'Practical and Tightly-Secure Digital Signatures and Authenticated Key Exchange'. In: *Advances in Cryptology – CRYPTO 2018*. Ed. by Hovav Shacham and Alexandra Boldyreva. Cham: Springer International Publishing, 2018, pp. 95–125.

[GJKW07] Eu-Jin Goh, Stanislaw Jarecki, Jonathan Katz and Nan Wang. 'Efficient Signature Schemes with Tight Reductions to the Diffie-Hellman Problems'. In: *Journal of Cryptology* 20.4 (2007), pp. 493–514.

[GMMV03] David Galindo, Sebastià Martín, Paz Morillo and Jorge L. Villar. 'Easy Verifiable Primitives and Practical Public Key Cryptosystems'. In: *Information Security*. Ed. by Colin Boyd and Wenbo Mao. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 69–83.

[GMR88] Shafi Goldwasser, Silvio Micali and Ronald L Rivest. 'A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks'. In: *SIAM Journal on Computing* 17.2 (1988), pp. 281–308.

[GPV08] Craig Gentry, Chris Peikert and Vinod Vaikuntanathan. 'Trapdoors for Hard Lattices and New Cryptographic Constructions'. In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. ACM. 2008, pp. 197–206.

[GR06] Craig Gentry and Zulfikar Ramzan. 'Identity-Based Aggregate Signatures'. In: *Public Key Cryptography - PKC 2006*. Ed. by Moti Yung, Yevgeniy Dodis, Aggelos Kiayias and Tal Malkin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 257–273.

[HJ12] Dennis Hofheinz and Tibor Jager. 'Tightly Secure Signatures and Public-Key Encryption'. In: (2012), pp. 590–607. DOI: 10.1007/978-3-642-32009-5_35. URL: https://doi.org/10.1007/978-3-642-32009-5_35.

[HJK11a] Dennis Hofheinz, Tibor Jager and Eike Kiltz. 'Short Signatures from Weaker Assumptions'. In: *Advances in Cryptology – ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 647–666.

[HJK11b] Dennis Hofheinz, Tibor Jager and Eike Kiltz. 'Short Signatures from Weaker Assumptions'. In: *ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, Dec. 2011, pp. 647–666. DOI: 10.1007/978-3-642-25385-0_35.

[HJK12] Dennis Hofheinz, Tibor Jager and Edward Knapp. 'Waters Signatures with Optimal Security Reduction'. In: *Public Key Cryptography – PKC 2012*. Ed. by Marc Fischlin, Johannes Buchmann and Mark Manulis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 66–83.

[HJP18] Dennis Hofheinz, Dingding Jia and Jiaxin Pan. 'Identity-Based Encryption Tightly Secure Under Chosen-Ciphertext Attacks'. In: *Advances in Cryptology – ASIACRYPT 2018*. Ed. by Thomas Peyrin and Steven Galbraith. Cham: Springer International Publishing, 2018, pp. 190–220.

[HK12a] Dennis Hofheinz and Eike Kiltz. 'Programmable Hash Functions and Their Applications'. In: vol. 25. 3. 2012, pp. 484–527.

[HK12b] Dennis Hofheinz and Eike Kiltz. 'Programmable Hash Functions and Their Applications'. In: *Journal of Cryptology* 25.3 (2012), pp. 484–527.

[HKKKH17] Gunnar Hartung, Björn Kaidel, Alexander Koch, Jessica Koch and Dominik Hartmann. 'Practical and Robust Secure Logging from Fault-Tolerant Sequential Aggregate Signatures'. In: *Provable Security*. Ed. by Tatsuaki Okamoto, Yong Yu, Man Ho Au and Yannan Li. Cham: Springer International Publishing, 2017, pp. 87–106.

[HKKKR16] Gunnar Hartung, Björn Kaidel, Alexander Koch, Jessica Koch and Andy Rupp. 'Fault-Tolerant Aggregate Signatures'. In: *Public-Key Cryptography – PKC 2016*. Ed. by Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano and Bo-Yin Yang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 331–356.

[HKS15] Dennis Hofheinz, Jessica Koch and Christoph Striecks. 'Identity-Based Encryption with (Almost) Tight Security in the Multi-instance, Multi-ciphertext Setting'. In: *Public-Key Cryptography – PKC 2015*. Ed. by Jonathan Katz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 799–822.

[HKW15] Susan Hohenberger, Venkata Koppula and Brent Waters. 'Universal Signature Aggregators'. In: *Advances in Cryptology - EUROCRYPT 2015*. Ed. by Elisabeth Oswald and Marc Fischlin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 3–34.

[HL02] Jeremy Horwitz and Ben Lynn. 'Toward Hierarchical Identity-Based Encryption'. In: *Advances in Cryptology — EUROCRYPT 2002*. Ed. by Lars R. Knudsen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 466–481.

[HSW13] Susan Hohenberger, Amit Sahai and Brent Waters. 'Full Domain Hash from (Leveled) Multilinear Maps and Identity-Based Aggregate Signatures'. In: *Advances in Cryptology – CRYPTO 2013*. Ed. by Ran Canetti and Juan A. Garay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 494–512.

[HW09a] Susan Hohenberger and Brent Waters. 'Realizing Hash-and-Sign Signatures under Standard Assumptions'. In: *EUROCRYPT 2009*. Ed. by Antoine Joux. Vol. 5479. LNCS. Springer, Apr. 2009, pp. 333–350. DOI: 10.1007/978-3-642-01001-9_19.

[HW09b] Susan Hohenberger and Brent Waters. 'Short and Stateless Signatures from the RSA Assumption'. In: *Advances in Cryptology - CRYPTO 2009*. Ed. by Shai Halevi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 654–670.

[HW09c] Susan Hohenberger and Brent Waters. 'Short and Stateless Signatures from the RSA Assumption'. In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Aug. 2009, pp. 654–670. DOI: 10.1007/978-3-642-03356-8_38.

[Jag12] Tibor Jager. *Digitale Signaturen*. Lecture Notes. Online erhältlich unter https://www.tiborjager.de/DigitaleSignaturen.pdf; abgerufen am 10. September 2018. 2012.

[Kat10] Jonathan Katz. *Digital signatures*. Springer Science & Business Media, 2010.

[KOF17] Kaisei Kajita, Kazuto Ogawa and Eiichiro Fujisaki. 'A Constant-Size Signature Scheme with Tighter Reduction from CDH Assumption'. In: *Information Security*. Ed. by Phong Q. Nguyen and Jianying Zhou. Cham: Springer International Publishing, 2017, pp. 137–154.

[KR00] Hugo Krawczyk and Tal Rabin. 'Chameleon Signatures'. In: *NDSS 2000*. Feb. 2000.

[KRS99] Ravi Kumar, Sridhar Rajagopalan and Amit Sahai. 'Coding Constructions for Blacklisting Problems without Computational Assumptions'. In: *CRYPTO'99*. Ed. by Michael J. Wiener. Vol. 1666. LNCS. Springer, Aug. 1999, pp. 609–623. DOI: 10.1007/3-540-48405-1_38.

[KS64] W Kautz and Roy Singleton. 'Nonrandom Binary Superimposed Codes'. In: *IEEE Transactions on Information Theory* 10.4 (1964), pp. 363–377.

[KW03] Jonathan Katz and Nan Wang. 'Efficiency Improvements for Signature Schemes with Tight Security Reductions'. In: *Proceedings of the 10th ACM Conference on Computer and Communications Security*. ACM. 2003, pp. 155–164.

[Lam79] Leslie Lamport. *Constructing Digital Signatures from a One-Way Function*. Tech. rep. Technical Report CSL-98, SRI International Palo Alto, 1979.

[LJYP14] Benoît Libert, Marc Joye, Moti Yung and Thomas Peters. 'Concise Multi-challenge CCA-Secure Encryption and Signatures with Almost Tight Security'. In: *Advances in Cryptology – ASIACRYPT 2014*. Ed. by Palash Sarkar and Tetsu Iwata. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–21.

[LLY13] Kwangsu Lee, Dong Hoon Lee and Moti Yung. 'Sequential Aggregate Signatures with Short Public Keys: Design, Analysis and Implementation Studies'. In: *Public-Key Cryptography – PKC 2013*. Ed. by Kaoru Kurosawa and Goichiro Hanaoka. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 423–442.

[LMRS04a] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin and Hovav Shacham. 'Sequential Aggregate Signatures from Trapdoor Permutations'. In: *Advances in Cryptology - EUROCRYPT 2004*. Ed. by Christian Cachin and Jan L. Camenisch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 74–90.

[LMRS04b] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin and Hovav Shacham. 'Sequential Aggregate Signatures from Trapdoor Permutations'. In: *EUROCRYPT 2004*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. LNCS. Springer, May 2004, pp. 74–90. DOI: 10.1007/978-3-540-24676-3_5.

[LOSSW06] Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham and Brent Waters. 'Sequential Aggregate Signatures and Multisignatures Without Random Oracles'. In: *EUROCRYPT 2006*. Ed. by Serge Vaudenay. Vol. 4004. LNCS. Springer, 2006, pp. 465–485. DOI: 10.1007/11761679_28.

[LVRW06] PC Li, GHJ Van Rees and R Wei. 'Constructions of 2-Cover-Free Families and Related Separating Hash Families'. In: *Journal of Combinatorial Designs* 14.6 (2006), pp. 423–440.

[LYWY17] Song Luo, Lu Yan, Jian Weng and Zheng Yang. 'New Proof for BKP IBE Scheme and Improvement in the MIMC Setting'. In: *Information Security Practice and Experience*. Ed. by Joseph K. Liu and Pierangela Samarati. Cham: Springer International Publishing, 2017, pp. 136–155.

[MH78] Ralph Merkle and Martin Hellman. 'Hiding Information and Signatures in Trapdoor Knapsacks'. In: *IEEE transactions on Information Theory* 24.5 (1978), pp. 525–530.

[MT08] Di Ma and Gene Tsudik. *A New Approach to Secure Logging*. Cryptology ePrint Archive, Report 2008/185. http://eprint.iacr.org/2008/185. 2008.

[Nev08] Gregory Neven. 'Efficient Sequential Aggregate Signed Data'. In: *Advances in Cryptology – EUROCRYPT 2008*. Ed. by Nigel Smart. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 52–69.

[NR04] Moni Naor and Omer Reingold. 'Number-Theoretic Constructions of Efficient Pseudo-Random Functions'. In: *Journal of the ACM (JACM)* 51.2 (2004), pp. 231–262.

[NY89] Moni Naor and Moti Yung. 'Universal One-Way Hash Functions and their Cryptographic Applications'. In: *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*. ACM. 1989, pp. 33–43.

[Rab79] Michael O Rabin. *Digitalized Signatures and Public-Key Functions as Intractable as Factorization*. Tech. rep. MASSACHUSETTS INST OF TECH CAMBRIDGE LAB FOR COMPUTER SCIENCE, 1979.

[Rom90] John Rompel. 'One-Way Functions are Necessary and Sufficient for Secure Signatures'. In: *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*. ACM. 1990, pp. 387–394.

[RS09] Markus Rückert and Dominique Schröder. 'Aggregate and Verifiably Encrypted Signatures from Multilinear Maps without Random Oracles'. In: *Advances in Information Security and Assurance*. Ed. by Jong Hyuk Park, Hsiao-Hwa Chen, Mohammed Atiquzzaman, Changhoon Lee, Tai-hoon Kim and Sang-Soo Yeo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 750–759.

[RSA78] Ronald L Rivest, Adi Shamir and Leonard Adleman. 'A Method for Obtaining Digital Signatures and Public-Key Cryptosystems'. In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[Rus94] Miklós Ruszinkó. 'On the Upper Bound of the Size of the r-Cover-Free Families'. In: *Journal of Combinatorial Theory, Series A* 66.2 (1994), pp. 302–310.

[Sch11] Sven Schäge. 'Tight Proofs for Signature Schemes without Random Oracles'. In: *Advances in Cryptology – EUROCRYPT 2011*. Ed. by Kenneth G. Paterson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 189–206.

[Seu12] Yannick Seurin. 'On the Exact Security of Schnorr-Type Signatures in the Random Oracle Model'. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 554–571.

[Sha84] Adi Shamir. 'Identity-Based Cryptosystems and Signature Schemes'. In: *Workshop on the Theory and Application of Cryptographic Techniques*. Springer. 1984, pp. 47–53.

[Str15] Christoph Striecks. 'On Cryptographic Building Blocks and Transformations'. PhD thesis. Karlsruhe Institute of Technology, 2015. URL: `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000050195`.

[SW05] Amit Sahai and Brent Waters. 'Fuzzy Identity-Based Encryption'. In: *Advances in Cryptology – EUROCRYPT 2005*. Ed. by Ronald Cramer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 457–473.

[Wat05] Brent Waters. 'Efficient Identity-Based Encryption Without Random Oracles'. In: *Advances in Cryptology – EUROCRYPT 2005*. Ed. by Ronald Cramer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 114–127.

[Wat09] Brent Waters. 'Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions'. In: *Advances in Cryptology - CRYPTO 2009*. Ed. by Shai Halevi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 619–636.

[WCD18] Gaoli Wang, Zhenfu Cao and Xiaolei Dong. 'Improved Fault-Tolerant Aggregate Signatures'. In: *The Computer Journal* (2018).

[XQZL14] Lu Xiuhua, Wen Qiaoyan, Jin Zhengping and Wang Licheng. 'A Lattice-Based Tag-Based Signature Scheme with Shorter Signature Length in the Standard Model'. In: (2014).