On Cryptographic Building Blocks and Transformations

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

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

genehmigte

Dissertation

von

Christoph Striecks

aus Salzwedel

Tag der mündlichen Prüfung: 26.06.2015

Erster Gutachter: Jun.-Prof. Dr. Dennis Hofheinz

Zweiter Gutachter: Prof. Dr. Eike Kiltz

Contents

1 Introduction **19**

1.1 Abstract of Result 1: Confined Guessing 21

1.2 Abstract of Result 2: (Almost) Tight IBE Security 23

1.3 Abstract of Result 3: A Generic View on Trace-and-Revoke Systems 25

1.4 Outline 27

2 Preliminaries **29**

3 Confined Guessing **37**

3.1 Preliminaries 43

3.2 (Mildly-Secure) Tag-Based Signatures 44

3.3 From Mild to Distinct-Message Non-Adaptive Security 46

3.4 From Distinct-Message Non-Adaptive to Full Security 50

3.5 With a View to Fully Secure Signature Instantiations 53

4 (Almost) Tight IBE Security **55**

4.1 Extended Nested Dual System Groups 62

4.2 An (Almost) Tightly Secure IBE 66

5 A Generic View on Trace-and-Revoke Systems 87

5.1 Preliminaries 93

5.2 An EDDH-based TEHPS Instance 95

5.3 Traceability of an EDDH-based RKEM 99

5.3.1 $(1, 2/3)$-Sid-traceability of RKEM 102

5.3.2 $((t+1)/2, \varepsilon)$-Sid-traceability of RKEM 108

5.3.3 Potential Generalizations of Our Tracing Result 112

6 Conclusion and Open Problems 115

List of Figures

2.1 EUF-CMA-experiment for DS schemes. 31

2.2 Non-adaptive t-RKEM-IND-CPA experiment for RKEMs. . . 32

3.1 EUF-dnaCMA-experiment for DS schemes. 43

3.2 The m-EUF-naCMA experiment for tag-based signature schemes. 45

3.3 An EUF-dnaCMA-secure signature scheme. 47

3.4 Schematic representation of (non-aborting) A_1. 49

3.5 An EUF-CMA-secure signature scheme. 51

3.6 Schematic representation of (non-aborting) A_1. 52

3.7 Schematic representation of (non-aborting) A_2. 53

4.1 Schematic description of the (μ, q)-IBE-IND-CPA-security experiment. 59

4.2 (μ, q)-IBE-IND-CPA experiment for IBE schemes. 67

5.1 Assumptions used in this chapter and their connections. The arrows indicate that the EDDH assumption is implied by the DDH and DCR assumptions, and that the DDH assumption implies the CDH assumption 90

5.2 Security experiments for (sid-)traceability of an RKEM. 101

Own Publications

1. Dennis Hofheinz, 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.

2. Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, and Christoph Striecks. Confined Guessing: New Signatures from Standard Assumptions. Journal of Cryptology, 28(1):176–208, January 2015.

3. Dennis Hofheinz and Christoph Striecks. A Generic View on Trace-and-Revoke Broadcast Encryption Schemes. In Josh Benaloh, editor, CT-RSA 2014, pages 48–63. Springer, February 2014.

4. Eduarda S. V. Freire, Dennis Hofheinz, Kenneth G. Paterson, and Christoph Striecks. Programmable Hash Functions in the Multilinear Setting. In Ran Canetti and Juan A. Garay, editors, CRYPTO 2013, pages 513–530. Springer, August 2013.

5. Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, Jae Hong Seo, and Christoph Striecks. Practical Signatures From Standard Assumptions. In Thomas Johansson and Phong Q. Nguyen, editors, EUROCRYPT 2013, pages 461–485. Springer, May 2013.

Acknowledgments

First of all, I want to thank my supervisor Jun.-Prof. Dennis Hofheinz for his broad support and guidance over the time. Without his rich cryptographic intuition and personal sense, this work would not have been possible. I am deeply grateful to have worked with him while writing this thesis. Further, I wish to thank Prof. Eike Kiltz. I feel honored that he agreed to be the second thesis referee. To continue, I want to thank Prof. Jörn Müller-Quade, Dr. Willi Geiselmann, and Carmen Manietta for their support over the years at the KIT. During the time, I had the chance to meet and to work with a lot of cryptographers. I especially wish to thank my co-authors Prof. Kenny Paterson, Prof. Jae Hong Seo, Dr. Florian Böhl, Dr. Eduarda Freire, Dr. Tibor Jager, and Jessica Koch as well as my group colleagues Dr. Andy Rupp and Julia Hesse. Further, I want to thank all members in the ITI crypto group of Prof. Jörn Müller-Quade. Lastly, I am very very glad for the extensive and permanently support by my family and friends$^{\diamondsuit}$.

Abstract

Cryptographic building blocks play a central role in cryptography. For instance, there exist well-established definitions of the cryptographic building blocks encryption and (digital) signatures as well as standard security notions thereof. Further, cryptographic building blocks might be constructed modularly, i.e., emerge out of other cryptographic building (sub-)blocks. To argue about security in an appropriate model, one usually transforms any efficient adversary on the emerged building block to an efficient adversary on the underlying building block(s). Thus, assuming that the underlying building blocks are secure, the emerged building block must be secure. Essentially, one reduces the security of the emerged building block to the security of the underlying building blocks. This form of reduction is an example of a cryptographic transformation and can be analyzed, e.g., in terms of efficiency and *tightness*. (Cryptographic transformations are omnipresent in (modern) cryptography.) In this thesis, we consider cryptographic building blocks and transformations in the following three aspects:

Confined Guessing. We give a cryptographic transformation to construct secure digital signatures. First, we define a new signature building block, dubbed tag-based signatures (where a signature additionally holds a tag from a specific tag space), together with a milder security definition. The intention is that this milder form of security might be easier to achieve when instantiating (concrete) tag-based signatures. Secondly, from several mildly secure tag-based signature instances, we derive digital signatures with compact parameters in the (even stronger) standard security setting via a cryptographic transformation. Intermediately, we use a technique called *confined guessing*, where we partition the tag space in the transformation such that there exists a large enough (but not too large) "confined" partition which allows us to guess tags with significant probability. Further, the concept of confined

guessing gives rise to instantiate new efficient and compact standard-model[1] secure signature. However, the instantiations are not part of this thesis and we solely focus on the confined guessing technique. The discussed concept was published in [BHJ+13, BHJ+15].

(Almost) Tight IBE Security. Consider an "extended" cryptographic building block, named identity-based encryption (IBE), where the encryption algorithm only needs the recipient's public ID (e.g., e-mail) and message, besides some public system parameters. Common IBE security notions only deal with one instance and one ciphertext, but real-world scenarios often involve multiple instances with multiple ciphertexts. However, we can trivially transform one-instance, one-ciphertext security to multi-instance, multi-ciphertext security; unfortunately, in this case, the security guarantees degrade in the number of instances and ciphertexts. (This often leads to more expensive computations in practice when implementing the systems and to the required a-priory knowledge of the number of instances and ciphertexts.) We say, an IBE scheme is (almost) tightly secure if its security guarantees *do not* degrade in the number of instances, users per instance, or ciphertexts. At Crypto 2013, Wee and Chen proposed the first (almost) tightly secure IBE in the one-instance, one-ciphertext security setting under a simple assumption[2]. We extend their underlying cryptographic building block such that we are able to obtain an efficient (almost) tight reduction from the security of our IBE to the security of the extended underlying cryptographic building block in the (even stronger) multi-instance, multi-ciphertext security setting. Concretely, we give an (almost) tight cryptographic transformation from any efficient adversary on our IBE in the multi-instance, multi-ciphertext security setting to an efficient adversary on the underlying building block. This approach was published in [HKS15].

A Generic View on Trace-and-Revoke Systems. Trace-and-revoke building blocks are for instance used in content protection (e.g., pay-TV). The idea is that only privileged (i.e., non-revoked) users can decrypt ciphertexts while, additionally, malicious users — who share their secret information with others — can be traced and afterwards excluded (i.e., revoked) from the system. We give a generic trace-and-revoke instantiation view by, first,

[1] The standard model is an established computational model in cryptography.

[2] We define simple assumptions in the preliminaries. Loosely speaking, a simple assumption is at least independent of adversarially queries.

extending a generic work from Wee (Eurocrypt 2011) concerning the revocation techniques. Concretely, Wee gives an underlying building block that is used to construct revocation schemes. Essentially, we connect a generic assumption[3] with the work from Wee. (This yields a slightly different generic view on revocation systems.) In the second part, we show the tracing capability of those revocation systems by picking up on and extending established techniques. (We mention that for some of Wee's revocation instantiations, it is not known if they are traceable.) Put together, we derive a new generic view of trace-and-revoke systems that generalizes known and new trace-and-revoke instantiations under simple assumptions. This work was published in [HS14].

[3]Loosely speaking, a generic assumption generalizes more concrete assumptions.

Zusammenfassung

Kryptographische Bausteine und Transformationen sind ein fester Bestandteil der modernen Kryptographie. In der einschlägigen Literatur sind kryptographische Bausteine, wie beispielsweise Verschlüsselung oder (digitale) Signaturen, allgegenwärtig. Zudem existieren passende Sicherheitsdefinitionen. Kryptographische Bausteine können wiederum aus anderen kryptographischen Bausteinen zusammengesetzt sein. Diese Modularität erlaubt das Betrachten von kryptographischen Systemen als Zusammensetzung anderer kryptographischer Bausteine. Um Sicherheitsgarantien für den zusammengesetzten Baustein zu geben, kann eine kryptographische Transformation benutzt werden. Diese garantiert in der Regel, dass jeder Angreifer auf den zusammengesetzten Baustein einen Angreifer auf die darunterliegende Bausteine impliziert. Wenn wir annehmen, dass die darunterliegenden Bausteine sicher sind, dann muss der daraus zusammengesetzte Baustein ebenfalls sicher sein. Damit wurde die Sicherheit des zusammengesetzten Bausteins auf die Sicherheit der darunterliegenden Bausteine reduziert. Diese Art von Reduktion ist ein Beispiel für eine kryptographische Transformation und kann in Hinblick auf Effizienz und Schärfe (engl. tightness) untersucht werden. In der vorliegenden Arbeit werden drei Ergebnisse im Rahmen von kryptographischen Bausteinen und Transformationen vorgestellt:

Confined Guessing. Wir geben eine kryptographische Transformation an, die es erlaubt, sichere digitale Signaturen zu konstruieren In einem Zwischenschritt definieren wir einen neuen kryptographischen Baustein, der als tag-basierte digitale Signaturen bezeichnet wird. (Eine Signatur in diesem Sinne besteht zusätzlich aus einem Tag (Etikett).) Zudem wird eine abgeschwächte, aber dennoch einsichtige Sicherheitsdefinition für tag-basierte Signaturen gegeben. Die Idee dahinter ist, dass eine abgeschwächte Sicherheitsdefinition es oftmals erlaubt, effiziente Instanziierungen zu konstruieren. Weit-

erhin wird eine kryptographische Transformation beschrieben, die mehrere abgeschwächt sichere tag-basierte Signaturen parallel nutzt, um sichere digitale Signaturen mit kompakten Parametern zu konstruieren. Dabei nutzt die Transformation das Konzept *confined guessing*. Dieses Konzept teilt die Menge aller Tags in disjunkte Teilmengen auf und findet eine *begrenzte* Teilmenge, die groß genug (aber nicht zu groß) ist, sodass Tags mit nicht-vernachlässigbarer Wahrscheinlichkeit geraten werden können. Dies ist vonnöten, um eine effiziente Transformation zu zeigen. Das Konzept confined guessing erlaubt zudem, sehr effiziente sicher Instanziierungen von digilen Signaturen im Standardmodell[4] anzugeben. Diese Instanziierungen sind jedoch nicht Teil der Arbeit, es wird ausschließlich auf die Transformation in Zusammenhang mit confined guessing eingegangen. Dieses Konzept wurde in [BHJ+13, BHJ+15] veröffentlicht.

(Fast) scharfe IBE-Sicherheit. Identitäts-basierte Verschlüsselung (IBE) erlaubt es, eine verschlüsselte Nachricht (Chiffrat) an einen Empfänger zu senden, von dem nur ein öffentlicher Identifikator (z. B. die E-Mail-Adresse) bekannt sein muss. Geläufige IBE-Sicherheitsdefinitionen betrachten nur eine IBE-Instanz und ein Chiffrat, wobei in realistischen Szenarien durchaus mehrere Instanzen mit mehreren Chiffraten vorkommen können. Es ist bekannt, dass die Sicherheit im Eine-Instanz-ein-Chiffrat-Szenario die Sicherheit im Mehrere-Instanzen-mehrere-Chiffrate-Szenario impliziert. Allerdings garantiert diese Transformation nur eine abgeschwächte Sicherheit im Vergleich zur Sicherheit im Eine-Instanz-ein-Chiffrat-Szenario. Die Folge ist, dass bei der Implementierung in der Praxis oftmals größere Schlüssellängen und damit intensivere Berechnungen vonnöten sind, um ein ähnliches Sicherheitsniveau zum Eine-Instanz-ein-Chiffrat-Szenario zu garantieren. Zudem müssen die Anzahl der Instanzen und die Anzahl der Chiffrate vor dem Aufsetzen des Systems bekannt sein. Ein IBE-System ist (fast) scharf sicher, wenn dessen Sicherheitsgarantien nicht von der Anzahl der Instanzen, Nutzer pro Instanz oder Anzahl der Chiffrate abhängen. Auf der Crypto-2013-Konferenz stellten Chen und Wee das erste (fast) scharf sichere IBE-System im Eine-Instanz-ein-Chiffrat-Szenario vor (das auf einer einfachen[5] Annahme basiert). Wir erweitern deren zugrundeliegenden Baustein und geben ein (fast) scharf sicheres IBE-System im Mehrere-Instanzen-mehrere-Chiffrate-

[4]Das Standardmodell ist ein gängiges Berechnungsmodell in der Kryptographie.

[5]Wir definieren einfache Annahmen in Kapitel 2. Unter anderem hängen einfache Annahmen nicht von angreifer-spezifischen Anfragen ab.

Szenario an. Genauer gesagt, wir reduzieren die Sicherheit unseres IBE-Systems auf die Sicherheit des zugrundeliegenden Bausteins. Dabei zeigen wir, dass diese Transformation im Mehrere-Instanzen-mehrere-Chiffrate-Szenario (fast) scharf ist. Dieser Ansatz wurde in [HKS15] publiziert.

Eine generische Sicht auf Trace-and-Revoke-Systeme. Trace-and-Revoke-Syteme finden Anwendung im Bereich der digitalen Rechteverwaltung. Diese Systeme erlauben es, nur privilegierten Nutzern das Entschlüsseln von geschützen Inhalten zu ermöglichen. Gleichzeitig wird das Finden von Nutzern erlaubt, die ihr geheimes Schlüsselmaterial unbefugt an Dritte weitergeben. Wird ein Nutzer dessen überführt, kann dieser vom System ausgeschlossen werden und ist somit nicht mehr in der Lage, die geschützten Inhalte zu entschlüsseln. (Diese böswilligen Nutzer werden Verräter (Traitors) genannt, das Verfahren wird als Tracing bezeichnet.) Wir geben eine neue Instanziierung eines kryptographischen Bausteins an, der von Wee auf der Eurocrypt 2011 vorgestellt wurde. Weiterhin kann dieser Baustein genutzt werden, um Revoke-Systeme (vorerst ohne Tracinggarantien) zu konstruieren. Damit erhalten wir eine neue Sicht auf diese Art von Systemen unter einer generische Annahme[6], die von Hemenway und Ostrovsky auf der PKC 2012 vorgestellt wurde. Weiterhin zeigen wir, dass diese Revoke-Systeme das Tracing von Verrätern unterstützen. Zusammengefasst erhalten wir damit eine neue generische Sicht auf Trace-and-revoke-Systeme, die bestehende und neue Trace-and-revoke-Instanziierungen generalisieren. Der beschriebene Ansatz wurde in [HS14] veröffentlicht.

[6]Eine generische Annahme abstrahiert, vereinfacht gesagt, konkretere Annahmen.

Chapter 1

Introduction

The Digital World and Cryptography. The digital world is rapidly growing with all its benefits, e.g., faster communication or global interconnectivity. However, besides the benefits, this development also comes with significant risks, e.g., vulnerabilities of computer systems, data breaches, or surveillance. Tackling those risks is an important necessity and cryptography can be used to address many specific risks in the digital world. In today's life, cryptography and its applications are omnipresent. For example, online banking or the procedure of a log-in into an email account often utilizes certain functionalities from the domain of cryptography for a secure payment or a secure authentication, respectively. One important subdomain of cryptography is encryption that is used to prevent eavesdropping; for example, one can encrypt big portions of data (e.g., in databases) such that only privileged users are able to read the content. Another more evolved example is encrypted e-mail, where legitimate users are able to communicate securely and privately. Further subdomains of cryptography consider, e.g., digital signatures, data integrity, or authentication. All in all, cryptography is a broad field with a lot of applications in the digital world. A central question is: how to build solid cryptographic functionalities with certain properties (e.g., security guarantees)?

Cryptographic Building Blocks and Transformations. Developing cryptographic functionalities and, in general, cryptographic systems is highly non-trivial due to their often complex and interwoven structure. A well-known approach is to build such systems out of smaller components, called

cryptographic building blocks, and combine these building blocks in a modular way to form a more complex cryptographic block or system. The procedure of combining cryptographic building blocks is also a non-trivial task. However, this approach is appealing due to a modular view of the complex system and the idea that smaller components are often easier to understand. (Additionally, maintaining or exchanging small components instead of an entire complex system might be easier as well.) There are well-established cryptographic building blocks in the cryptographic literature, e.g., encryption or digital signatures with appropriate definitions of security. (The security of a cryptographic building block has to be defined carefully.) As indicated before, these building blocks are often combined to obtain a more evolved building block (or, more evolved system). To argue about security, one usually cryptographically transforms any malicious attacker (called "adversary") on the evolved building block or system into an attacker on the underlying building block(s). Put differently, assuming that the underlying cryptographic building block(s) are secure, the more evolved building block or system must be secure as well. This is an example for a cryptographic transformation and we can analyze such transformations regarding its efficiency and tightness.

Content and Results of this Thesis. This thesis will focus on cryptographic building blocks and transformations in the cryptographic subdomains digital signatures and encryption — where in the encryption case we will particularly study identity-based encryption (IBE) and broadcast encryption (BE). Digital signatures, IBE, and BE are important cryptographic areas with a large body of research in the cryptographic community. Digital signatures are for example used in the well-known Transport Layer Security (TLS) protocol [DR06] over the Internet or in operating system updates. The idea of IBE is to simplify certificate management. Certificate management is widely used in practice and its deployment is usually very costly. Interestingly, digital signatures and IBE are closely related; that is, an IBE system can easily be converted into a digital signature system [BF03]. The essence of BE is the secure and efficient distribution of sensitive data to a wider audience which, for example, has application in pay-TV. This thesis presents three new results in the mentioned cryptographic areas. Particularly, we will focus on building blocks and examine cryptographic transformations within digital signatures, IBE, and BE. Concretely:

1. We will describe a new transformation that uses a technique called

confined guessing in the realm of digital signatures.

2. We present a new transformation in the IBE setting and extend an underlying cryptographic building block.

3. We develop a new (generic) instantiation view of a cryptographic building block in the BE setting.

Further, we will start with an introduction and a general overview of all three results in Section 1.1, Section 1.2, and Section 1.3.

1.1 Abstract of Result 1: Confined Guessing

Digital Signatures. Digital signatures are cryptographic building blocks that can be seen as a digital counterpart to handwritten signatures and are used widely in today's digitally connected world, e.g., in the TLS protocol in combination with HTTPS over the Internet. In such digital signature systems, given a security parameter[1], a user is able to generate a verification and secret key pair. The verification key is usually published and publicly available while the secret key is kept, as the name already suggests, secret. This secret key, which for example can be embedded in a smart card, is used to sign a message (e.g., an e-mail). Both, the signature and the message can then be transmitted, e.g., over the Internet. To verify a signature on a message, one uses the user's verification key to validate the signature on the message. The correctness and the security properties of digital signature schemes usually guarantee integrity, authenticity, and unforgeability. Integrity basically ensures that the message was not altered during transfer. Authenticity stipulates that the signature was created by the user who claims to be the creator of the signature on some message. Unforgeability says that it is difficult to create a new valid message-signature pair without knowing the corresponding secret key.

Efficiency, Security, Cryptographic Transformations (e.g., Reductions) in the Digital Signature Context. There are various flavors

[1]Basically, the security parameter is given to any party using (and also abusing) the system. Further, all computational power within the (standard) model and the success probabilities depend on the security parameter.

of digital signature schemes in the cryptographic literature, e.g., mainly in terms of efficiency and security. Concerning efficiency, the goal is to construct digital signature schemes with "short" parameters, i.e., short key and signature sizes. Concerning security, one particular notion of digital signature security, dubbed EUF-CMA-security, is considered to be the standard security notion that should be achieved by digital signature schemes. Loosely speaking, a signature scheme is EUF-CMA-secure, if no efficient adversary is able to forge a signature on a new message except with negligible probability[2]. Hence, the goal is to find EUF-CMA-secure digital signature schemes with short parameters.

To argue about concrete[3] security guarantees, one usually efficiently reduces (or transforms) the EUF-CMA-security of a digital signature scheme to the hardness of a computational problem. Put differently, one usually shows that any efficient adversary on the signature scheme yields an efficient problem solver. This contradicts the assumption that the underlying problem is hard to solve and, hence, the digital signature scheme must be EUF-CMA-secure. (To make it clear, the described concept is also a form of cryptographic transformation or reduction.)

The Problem. Within the reduction in the EUF-CMA-context, the adversary at some point outputs a signature and a message (for which it does not know the secret key). This can be seen as an attempt to forge a valid signature-message pair. Concerning the reduction, this message should be somehow connected to the computational problem. (Otherwise, loosely speaking, the adversary would not be of "any help.") Naively, the reduction would try to embed the computational problem instance into the system's parameters and "connect" it to some message the adversary hopefully will output. Unfortunately, there are usually too many possibilities for the adversary to choose a forgery message. Hence, in this naive approach, the reduction would not be able to guess efficiently which message is going to be output by the adversary. (We mention that we are only interested in efficient transformations.) Reduction strategies to overcome this problem are known in the cryptographic literature, e.g., partitioning. (See chapter 3 for

[2]To be concrete, a negligible probability is smaller than any inverse of a polynomial in the security parameter.

[3]Here, concrete means that we base the security of a signature scheme on a well-defined computational hard problem (i.e., which is assumed to be hard to solve efficiently).

more details.) However, some digital signature instantiations shown EUF-CMA-secure under the well-established computational Diffie-Hellman (CDH) assumption using the partitioning strategy suffer from relatively large verification keys.

Our Contribution. We present a different reduction or cryptographic transformation strategy, dubbed confined guessing, and we develop it two-staged. First, we introduce a new building block, named tag-based digital signatures (where a signature additionally carries a tag from tag space), together with a milder notion of security in comparison to EUF-CMA-security. This milder security notion can easier be achieved by tag-based signature instantiations. Secondly, we give a cryptographic transformation from the mildly secure tag-based signatures to an EUF-CMA-secure digital signatures. Within the transformation, we use the concept confined guessing, where we partition the tag space such that there exists a large enough (but not too large) "confined" partition. This approach allows us to guess tags with significant probability which, in turn, yields an efficient transformation. (We stress that we are only interested in efficient transformations.) All in all, this strategy gives rise to efficient EUF-CMA-secure digital signature instantiations (under concrete simple assumptions), e.g., an EUF-CMA-secure digital signature scheme under the computational Diffie-Hellman (CDH) assumption with compact verification keys. However, this thesis only covers the main strategy of confined guessing while the mentioned digital signature instantiations are not part of this work. Confined guessing was published in [BHJ+13, BHJ+15].

1.2 Abstract of Result 2: (Almost) Tight IBE Security

Identity-based Encryption. In an identity-based encryption (IBE) system, one can send an encrypted message by only using a recipient's public identifier string (e.g., the MAC or e-mail address). This is different to ordinary encryption schemes, where one usually needs some public key of the recipient to send encrypted messages. Initially, after receiving an encrypted message in the IBE system, the recipient obtains a user secret key from

some trustworthy authority and is henceforth able to decrypt the encrypted message (by using this user secret key). The idea is to simplify certificate management which is usually very costly in public-key infrastructures. However, despite the appealing idea of using only a recipient's public identifier for encryption note that an IBE system makes use of a trustworthy authority which knows a master secret key that is used to generate all user secret keys in the system.

Finally and interestingly, we want to mention that IBE systems and digital signatures are cryptographically related, i.e., an IBE scheme can easily be translated into a digital signature scheme [BF03].

Tight Security Reductions in the IBE Context. Common cryptographic transformations in the IBE context are as follows: one efficiently reduces the security of the IBE scheme to the security of the underlying building block(s). Equivalently, one efficiently transforms any efficient adversary on the IBE scheme to an efficient adversary on the underlying building block(s). Technically, one has to make sure that the success probability of the IBE adversary is not degraded too much within the reduction, i.e., loosely speaking, the success probability of the adversary on the underlying blocks should be large enough. Further, one can quantify the amount of the degradation which is referred to as the *loss* of the reduction. Often, this reduction loss is connected to the efficiency of the instantiation (e.g., a smaller loss often translates to smaller key sizes of the implementations in practice.) Hence, the goal is to provide security reduction with a small or (even better) no loss. If the loss is linear in the security parameter, then we say that the transformation is almost tight; if the loss is a constant in the security parameter, then we say that the transformation is tight.

(Almost) Tight IBE Schemes. Common IBE security notions only deal with one instance and one ciphertext while real-world scenarios often consider multiple instances with multiple ciphertexts. It is known that one can trivially lift one-instance, one-ciphertext security to multi-instance, multi-ciphertext security; unfortunately, in this case, the security guarantees degrade in the number of instances and ciphertexts. Essentially, the loss of such a transformation depends on the number of instances and ciphertexts. (As mentioned before, this often leads to more expensive computations in practice. Addionally, it also leads to the required a-priory knowledge of the

number of instances and ciphertexts.) In [CW13], Chen and Wee construct the first IBE scheme with an (almost) tight reduction in the one-instance, one-ciphertext setting. A natural question is: is it possible to extend their result to a setting with multiple instances and multiple ciphertexts while the property of an (almost) tight reduction can be preserved?

Our Contribution. We answer the question from the last paragraph in the affirmative. Concretely, we construct an (almost) tightly secure IBE scheme in the multi-instance, multi-ciphertext setting. We first define an extended security notion for IBE system that deals with multiple instances and multiple ciphertexts. This notion is a natural extension of the standard IBE security notion. Secondly, we extend the underlying cryptographic building block proposed by Chen and Wee such that we are able to prove an IBE scheme (almost) tightly secure in the multi-instance, multi-ciphertext setting. Concretely, we give an (almost) tight reduction from the security of the IBE to the security of the underlying building block in the multi-instance, multi-ciphertext setting. This work was published in [HKS15].

1.3 Abstract of Result 3: A Generic View on Trace-and-Revoke Systems

Revocable Key Encapsulation Mechanism. Revocable key encapsulation mechanisms (RKEMs) can be used in the context of content protection (e.g., pay-TV). Basically, an RKEM allows to exclude non-privileged (e.g., non-paying) users from decrypting the encrypted content while all other system users are able to decrypt. To this end, the system generates a master public key and a master secret key. Users can join the system and each new user obtains a user secret key from the system (using the master secret key). The encapsulation works as follows: given an excluded (or, revoked) set of users, a "shared" key is encapsulated such that only the users who are not in the revoked set are able to decapsulate the shared key. In general, the shared key is used to encrypt some payload data (e.g., a video file) and after successful decapsulation, the shared key is used to decrypt the encrypted payload. (Note that the encrypted payload is only broadcasted once and can be received and decrypted by all users except those in the revoked set.)

In some cases, these revocable systems incorporate a threshold parameter that limits the number of users who might group together and share their user-secret-key material. Up to this threshold bound, an RKEM guarantees correctness and an appropriate form of security.

Trace-and-Revoke Systems. Speaking of colluding users, consider privileged (i.e., non-revoked) users who share their user secret keys with each other (or, with others) and build a decapsulation box out of this secret material. As an example, one can think of users who sell their user secret keys in a decapsulation box over the Internet. (Those users are often referred to as "pirates" or "traitors.") A natural question is: how to catch those traitors? Assume that one (e.g., the police) is somehow able to obtain such a pirated box. The approach would be that if a pirated box is found, then there should be an efficient algorithm that uncovers at least one traitor only by examining the input-output behavior of the box. This technique is known as traitor tracing. Further, one can think of combining tracing with RKEMs. This would yield a powerful cryptographic tool. Namely, after at least one traitor is found through tracing, this user can be revoked (i.e., loosely speaking, the corresponding user secret key is not useful any more). There is some evidence that the combination of tracing and RKEMs is nontrivial to realize. Nevertheless, schemes are proposed in the cryptographic literature which achieve both; those schemes are called trace-and-revoke systems.

In [Wee11], Wee gives a generic view of RKEMs. His generic systems can be instantiated under three different well-established computational assumptions. We extend his work in two directions.

Our Contribution. Our first result yields an extension of Wee's work [Wee11]. Concretely, we connect a generic assumption due to Hemenway and Ostrovsky [HO12] with a cryptographic building block proposed by Wee. This cryptographic building block is used by Wee to construct RKEMs. Hence, we derive new RKEMs under a generic assumption. (This yields a new slightly different generic view on revocation system.) In our second result, we show that those RKEMs under the generic assumption are traceable. We extend and pick up on established techniques. (We mention that for some of Wee's RKEM instantiations, it is not known if they are traceable.) In detail, we provide a tracing algorithm that takes the pirate box and outputs a traitor who contributed in building this box. All in all, we derive

a new generic view of trace-and-revoke systems that generalizes known and new trace-and-revoke instantiations. This work was published in [HS14].

1.4 Outline

Concerning the outline, in Chapter 2, we give the required preliminaries used within this thesis. Further, Chapter 3 describes the first result, i.e., describes the confined guessing concept and shows how to obtain EUF-CMA-secure digital signatures using this concept. In Chapter 4, we give (almost) tightly secure IBE systems in the multi-instance, multi-ciphertext setting. Chapter 5 shows a generic instantiation view on trace-and-revoke systems. Finally, Chapter 6 concludes this thesis and hints at open problems.

Chapter 2

Preliminaries

Notation. For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$, and let $k \in \mathbb{N}$ be the security parameter. For a finite set $\mathcal{S}$, we denote by $s \leftarrow \mathcal{S}$ the process of sampling s uniformly from $\mathcal{S}$. For an algorithm A, let $y \leftarrow A(k, x)$ be the process of running A on input k, x with access to uniformly random coins and assigning the result to y. (We may omit to mention the k-input explicitly and assume that all algorithms take k as input.) To make the random coins r explicit, we write $A(k, x; r)$. We say an algorithm A is probabilistic polynomial time (PPT) if the running time of A is polynomial in k. A function f is negligible if its absolute value is smaller than the inverse of any polynomial (i.e., if $\forall c \exists k_0 \forall k \geq k_0 : |f(k)| < 1/k^c$). Further, a function f is significant if its absolute value is larger or equal than the inverse of some polynomial (i.e., if $\exists c, k_0 \forall k \geq k_0 : |f(k)| \geq 1/k^c$). We may write $q = q(k)$ if we mean that the value q depends polynomially on k. We write vectors in bold font, e.g., $\mathbf{v} = (v_1, \ldots, v_n)$ for a vectors 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 $\mathbf{v} = (v_i)_i$ in this case.) 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 $j \in \mathbb{N}$, and $\mathbf{v}_{-i} := (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$, for $i \in [n]$, and $s^{\mathbf{v}} := (s^{v_1}, \ldots, s^{v_n})$. For two random variables X, Y, we denote with $\mathsf{SD}(X\,;\,Y)$ is the statistical distance of X and Y. We might also say that X and Y are ε-close if $\mathsf{SD}(X\,;\,Y) \leq \varepsilon$.

Digital Signatures. A digital signature (DS) scheme SIG with message space $\mathcal{M}$ consists of three PPT algorithms $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ as follows:

Key generation. $\mathsf{Gen}(k)$, on input security parameter k, outputs public paramater pp, and verification and secret keys (pp, vk, sk). (We assume that Sig and Ver have implicitly access to pp.)

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

Verification. $\mathsf{Ver}(vk, \sigma, M)$, on input vk, σ, and message M, outputs $b \in \{0, 1\}$.

We define correctness and security of a DS scheme:

Correctness. For all $k \in \mathbb{N}$, for all $(vk, sk) \leftarrow \mathsf{Gen}(k)$, for all $M \in \mathcal{M}$, for all $\sigma \leftarrow \mathsf{Ext}(sk, M)$, we have $\mathsf{Ver}(vk, \sigma, M) = 1$.

EUF-CMA security [GMR88]. We say a DS scheme SIG is existentially unforgeable under chosen-message attacks (EUF-CMA-secure) if and only if any PPT adversary A has only negligible advantage in the following security experiment. First, A receives an honestly generated verification key vk. During the experiment, A has access to a signature oracle $\mathsf{Sig}(sk, \cdot)$. (Where sk is the corresponding secret key to vk.) Eventually, A outputs a signature σ^* on a forgery message M^*. If $\mathsf{Ver}(vk, \sigma^*, M^*) = 1$ and A has never queried a signature for M^*, then the experiment outputs 1. More formally, we define the advantage function for an adversary A as

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

where the experiment $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},A}(k)$ is given in Figure 2.1 and SIG is a DS scheme. Then we say SIG is EUF-CMA-secure if and only if for any PPT adversary A the function $\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},A}(k)$ is negligible in k.

Identity-Based Encryption. An identity-based encryption (IBE) scheme IBE with identity space $\mathcal{ID}$ and message space $\mathcal{M}$ consists of the following five PPT algorithms $(\mathsf{Par}, \mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$:

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

Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},A}(k)$
$(vk, sk) \leftarrow \mathsf{Gen}(k)$
$(M^*, \sigma^*) \leftarrow A^{\mathsf{Sig}(sk,\cdot)}(vk)$
if $\mathsf{Ver}(vk, M^*, \sigma^*) = 1$ and A has not queried $\mathsf{Sig}(sk, M^*)$
then return 1 else return 0

Figure 2.1: EUF-CMA-experiment for DS schemes.

Key generation. $\mathsf{Gen}(pp, sp)$, on input pp and sp, outputs master public and secret keys (mpk, msk).

Secret-key extraction. $\mathsf{Ext}(msk, id)$, on input msk and identity $id \in \mathcal{ID}$, outputs a user secret key usk_{id}.

Encryption. $\mathsf{Enc}(mpk, id, M)$, on input mpk, $id \in \mathcal{ID}$, and message $M \in \mathcal{M}$, outputs a ciphertext C.

Decryption. $\mathsf{Dec}(usk_{id}, C)$, on input usk_{id} and C, outputs $M \in \mathcal{M} \cup \{\perp\}$.

We define correctness in the following sense:

Correctness. For all $k, n = n(k) \in \mathbb{N}$, for all $(pp, sp) \leftarrow \mathsf{Par}(k, n)$, 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 $M \in \mathcal{M}$, for all $C \leftarrow \mathsf{Enc}(mpk, id, \mathcal{M})$, we have $\mathsf{Dec}(usk_{id}, C) = M$.

Security an IBE scheme is defined in chapter 4.

Revocable Key Encapsulation Mechanism. A revocable key encapsulation mechanism (RKEM) scheme with identity space $\mathcal{ID}$ and key space $\mathcal{K}$ consists of the following four PPT algorithms $(\mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$:

Key generation. $\mathsf{Gen}(k)$ outputs master public and secret keys (mpk, msk).

Secret-key extraction. $\mathsf{Ext}(msk, id)$, on input msk and identity $id \in \mathcal{ID}$, outputs a user secret key usk_{id}.

Encapsulation. $\mathsf{Enc}(mpk, \mathcal{R})$, on input mpk and revoked-identities set $\mathcal{R} \subseteq \mathcal{ID}$, outputs a ciphertext key pair (C, K).

Decapsulation. $\mathsf{Dec}(usk_{id}, C)$, on input usk_{id} and C, outputs $K \in \mathcal{K} \cup \{\perp\}$.

We define correctness and security of an RKEM in the following sense:

Correctness. For all $k \in \mathbb{N}$, for all $(mpk, msk) \leftarrow \mathsf{Gen}(k)$, for all $id \in \mathcal{ID}$, for all $usk_{id} \leftarrow \mathsf{Ext}(msk, id)$, for all $\mathcal{R} \subseteq \mathcal{ID}$, for all $(C, K) \leftarrow \mathsf{Enc}(mpk, \mathcal{R})$, we have $\mathsf{Dec}(usk_{id}, C) = K$.

Non-adaptive t-RKEM-IND-CPA security. We say an RKEM scheme is non-adaptively t-RKEM-IND-CPA-secure if and only if any PPT adversary A has only negligible advantage in the following security experiment. First, A outputs a revoked-identities set $\mathcal{R} \subseteq \mathcal{ID}$ and receives an honestly generated master public key mpk, user secret keys $usk_{id} \leftarrow \mathsf{Ext}(msk, id)$, for master secret key msk and identities all $id \in \mathcal{R}$, and a ciphertext-key pair (C, K_b), for $(C, K_0) \leftarrow \mathsf{Enc}(mpk, \mathcal{R}, M_b)$ and $K_1 \leftarrow \mathcal{K}$, for $b \leftarrow \{0, 1\}$. Eventually, A outputs a guess b'. Finally, if $b = b'$ and $|\mathcal{R}| \leq t$, then the experiment outputs 1.

More formally, we define the advantage function for an adversary A as

$$\mathsf{Adv}^{t\text{-RKEM-IND-CPA}}_{\mathsf{RKEM},A}(k) := |\Pr\left[\mathsf{Exp}^{t\text{-RKEM-IND-CPA}}_{\mathsf{RKEM},A}(k) = 1\right] - 1/2\,|,$$

where the experiment $\mathsf{Exp}^{t\text{-RKEM-IND-CPA}}_{\mathsf{RKEM},A}(k, n)$ is given in Figure 2.2 and RKEM is an RKEM. Then we say RKEM is non-adaptively t-RKEM-IND-CPA-secure if and only if the function $\mathsf{Adv}^{t\text{-RKEM-IND-CPA}}_{\mathsf{RKEM},A}(k)$ is negligible in k, for any PPT adversary A.

Experiment $\mathsf{Exp}^{t\text{-RKEM-IND-CPA}}_{\mathsf{RKEM},A}(k)$
- $\mathcal{R} \leftarrow A(k)$
- $(mpk, msk) \leftarrow \mathsf{Gen}(k)$
- $usk_{id} \leftarrow \mathsf{Ext}(msk, id)$, for all $id \in \mathcal{R}$
- $b \leftarrow \{0, 1\}$
- $(C, K_0) \leftarrow \mathsf{Enc}(mpk, \mathcal{R})$, $K_1 \leftarrow \mathcal{K}$
- $b' \leftarrow A(mpk, (usk_{id})_{id}, C, K_b)$
- if $b = b'$ and $|\mathcal{R}| \leq t$ then return 1 else return 0

Figure 2.2: Non-adaptive t-RKEM-IND-CPA experiment for RKEMs.

Pseudorandom Function Family. A pseudorandom function (PRF) family PRF with key space $\mathcal{K}$, domain $\mathcal{D}$, and range $\mathcal{R}$ is defined as $\mathsf{PRF} : \mathcal{K} \times \mathcal{D} \rightarrow$

$\mathcal{R},\ (K,X) \mapsto \mathsf{PRF}(K,X)$. (We also write $\mathsf{PRF}_K^{\mathcal{R}} : \mathcal{D} \to \mathcal{R},\ X \mapsto \mathsf{PRF}_K^{\mathcal{R}}(X)$, for $K \leftarrow \mathcal{K}$.) We say, for any PPT adversary D, the function

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF}_K^{\mathcal{R}},D}(k) := |\Pr\left[D^{\mathsf{PRF}_K^{\mathcal{R}}(\cdot)}(k)=1\right] - \Pr\left[D^{\mathsf{U}^{\mathcal{R}}(\cdot)}(k)=1\right]|$$

is negligible in k, for PRF $\mathsf{PRF}_K^{\mathcal{R}}$, for $K \leftarrow \mathcal{K}$, and truly random function $\mathsf{U}^{\mathcal{R}}$ to $\mathcal{R}$. (We refer to this as the indistinguishability property of a PRF.) Further, let $\mathsf{SampPRF}_{\mathcal{K},\mathcal{D},\mathcal{R}}$ be a PPT algorithm that samples PRF families for specific (efficiently sampleable) sets $\mathcal{K}, \mathcal{D}$, and $\mathcal{R}$ as defined above. Now, we can use the output of only one PRF $\mathsf{PRF}_K^{\mathcal{R}'}$, for (only one) $K \leftarrow \mathcal{K}$ and large enough domain $\mathcal{R}'$ (e.g., for $\mathcal{R}' = \{0,1\}^k$, for large enough k), as random coins for $\mathsf{SampPRF}_{\mathcal{K},\mathcal{D},\mathcal{R}_i}$ to derive several PRFs for arbitrary domains $(\mathcal{R}_i)_i$.

Pairing. Let G, H, and G_T be cyclic groups of known order written multiplicatively. A pairing $e : G \times H \to G_T$ has the following properties:

Bilinearity. $\forall a, b \in \mathbb{Z}, \forall g \in G, h \in H$, it holds that $e(g^a, h^b) = e(g,h)^{ab}$.

Non-degeneracy. For all generators $g \in G$ and all generators $h \in H$, it holds that $e(g,h) \neq 1$.

Efficiency. The map e is efficiently computable.

Lagrange Interpolation and Vandermonde Matrices. Fix a field $\mathbb{F}$ and $d+1$ values $x_0, \ldots, x_d \in \mathbb{F}$. The Vandermonde matrix $V_{x_0,\ldots,x_d} \in \mathbb{F}^{(d+1)\times(d+1)}$ is defined as

$$V_{x_0,\ldots,x_d} := \begin{pmatrix} 1 & x_0 & \ldots & x_0^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_d & \ldots & x_d^d \end{pmatrix}.$$

It is easy to see that $\det(V_{x_0,\ldots,x_d}) = \prod_{i<j}(x_j - x_i)$; in particular, $V_{x_0,\ldots,x_d}$ is invertible iff all x_i are distinct. We can evaluate a polynomial $f(x) = a_0 + a_1 x + \cdots + a_d x^d$ at $x_0, \ldots, x_d$ via

$$(f(x_0), f(x_1), \ldots, f(x_d))^\top = V_{x_0,\ldots,x_d} \cdot (a_0, a_1, \ldots, a_d)^\top.$$

Conversely, given values $y_0, \ldots, y_d \in \mathbb{F}$, we can via

$$(a_0, a_1, \ldots, a_d)^\top = V^{-1}_{x_0,\ldots,x_d} \cdot (y_0, y_1, \ldots, y_d)^\top$$

compute coefficients $a_0, \ldots, a_n \in \mathbb{F}$ of a polynomial $f(x) = a_0 + a_1 x + \cdots + a_d x^d$ such that $f(x_i) = y_i$. It will be useful to perform such matrix-vector multiplications "in the exponent," where generally a matrix $M = (M_{i,j}) \in \mathbb{F}^{n \times n}$ is known, and a vector $x = (x_i) \in \mathbb{F}^n$ is given in the form $X = (X_i) = (g^{x_i})$ for some g. We will write

$$M \circ X := (Y_1, \ldots, Y_n) \quad \text{with} \quad Y_i := \prod_{j=1}^{n} X_j^{M_{i,j}}.$$

If we write $y = (y_i)$ for the "exponent vector" with $Y_i = g^{y_i}$, this achieves $M \cdot x = y$.

Simple Assumptions. We restate the definition used in [HKS15]. A simple assumption is defined via a security game with an adversary. First, the adversary gets the computational problem instance (which only depends on the security parameter). Eventually, the adversary outputs a guess (without any previous interaction) and wins if the guess is a unique solution to the problem instance. A simple assumption stipulates that any PPT adversary only wins with negligible probability. (All assumptions considered in this work are simple in the above mentioned sense.)

Extended Decisional Diffie-Hellman (EDDH) Assumption. We restate the EDDH assumption defined by Hemenway and Ostrovsky [HO12]. Let $\mathcal{G}$ be a group and $\mathcal{K} \subset \mathbb{Z}$. For any PPT adversary A, the function

$$\mathsf{Adv}^{\mathsf{eddh}}_{A,\mathcal{G},G,H,\mathcal{K}}(k) := |\Pr\left[A(pars, g^{ab}) = 1\right] - \Pr\left[A(pars, g^{ab} \cdot h) = 1\right]|$$

is negligible in k, for $pars = (N, g, g^a, g^b)$, for subset $G \subset \mathcal{G}$, for subgroup $H \subseteq \mathcal{G}$, for H-group order N, for $g \leftarrow G$, for $h \leftarrow H$, and for exponents $a, b \leftarrow \mathcal{K}$. We assume that we can sample uniformly at random from G, H, and $\mathcal{K}$. Additionally, we define a PPT randomness extractor G such that $\mathsf{G}(h)$ with uniform $h \in H$ is pseudorandom.

Hemenway and Ostrovsky show that their EDDH assumption is implied by the decisional Diffie-Hellman (DDH) and the decisional composite residuosity (DCR) assumptions [HO12, Lemma 3 and Theorem 2].

Bilinear Decisional Diffie-Hellman (BDDH) Assumption. Let G, G_T be groups of prime order q and let $e : G \times G \longrightarrow G_T$ be a pairing. Further,

let $g \in G$ be a generator of G. For any PPT adversary D, we have that the function

$$\begin{aligned}\mathsf{Adv}_D^{\mathrm{bddh}}(k) := |\Pr\left[D(g, g^a, g^b, g^c, e(g,g)^{abc}) = 1\right] \\ - \Pr\left[D(g, g^a, g^b, g^c, e(g,g)^{z}) = 1\right]|\end{aligned}$$

is negligible in k, for (uniform) exponents $a, b, c, z \leftarrow \mathbb{Z}_q$.

Chapter 3

Confined Guessing

Digital Signatures. Loosely speaking, digital signature (DS) schemes are cryptographic building blocks that allow to use a secret key to sign a message, i.e., create a signature on a message such that the resulting signature together with the message can be verified under a verification key. The system guarantees integrity (i.e., the message was not altered during transfer) and authenticity (i.e., the message was signed with a secret key that corresponds to a specific verification key). The idea of digital signatures dates back at least to the work of Diffie and Hellman [DH76] and the first construction candidates were given in, e.g., [MH78, RSA78, Rab79].

(Efficient) Constructions of DS Schemes. DS constructions can be found in various research papers with a diversity of security and efficiency guarantees. Besides the already above mentioned schemes, there are (tree-based) signatures which can be constructed from (variants of) trapdoor permutations (e.g., [GMR88, BM88, BR93, CD95]) and from any one-way function (e.g., [Lam79, Rom90]), but all with rather large signatures. Other (tree-based) schemes are based on conventional encryption functions (e.g., [Mer88, Mer90]), on interactive protocols (e.g., [CD95]), or on pseudo-random functions and non-interactive zero-knowledge proofs (e.g., [BG90]). More (in many cases more efficient) constructions rely on specific hardness assumptions, e.g., some rely on the computational hardness of integer factorization and its variants (e.g., [GMY83, DN94, BR96]) or are based on the problem of computing discrete logarithms (e.g., [ElG84, Sch91, Oka93, PS96]) while some others are RSA-based (e.g., [CD96, CS99, HK08, HW09a, HW09b,

HJK11, BHJ+15]), Diffie-Hellman-based (e.g., [BLS01, BB04c, Wat05, HK08, HW09a, BHJ+13]), lattice-based (e.g., [GPV08, MP12, DM14, BHJ+15]), or d-LIN-based (e.g., [Wat09, Lew12, HJ12]).

(Standard) Security Notion for DS Schemes. Existentially unforgeability under chosen-message attacks (EUF-CMA) [GMR88] is considered to be the standard security notion for DS schemes. Usually, this notion is defined via an experiment with a PPT adversary on the DS scheme. Within this notion, the adversary receives an honestly generated verification key and is allowed to adaptively query signatures on messages. Eventually, it outputs a forgery, i.e., a forgery message together with a signature, and succeeds if the signature for this forgery message is correctly verified under the verification key. (Here, a forgery message is a message for which the adversary has not queried a signature before.) We say that an DS scheme is EUF-CMA-secure if any PPT adversary has only negligible advantage in winning the above experiment. It is almost redundant to say that there exist more security notions. However, many schemes (also many of the mentioned schemes above) yield this strong notion of security. (Further, we sometimes might write full security instead of EUF-CMA-security.)

Goal(s) in Constructing DS Schemes. One goal in constructing DS schemes is to provide EUF-CMA-secure stateless schemes with "short" parameters (i.e., the length of the signature, the verification and secret keys should be "short") under some reasonable assumption(s) in the standard model[1] with "efficient" key generation, signing, and verification. On a more generic level, one can also try to define (weaker-than-EUF-CMA) security notions which might be easier to achieve than EUF-CMA-security. To construct EUF-CMA-secure schemes, one then might try to "efficiently transform" the weaker variants into EUF-CMA-secure ones. (See [KR00] as an example for such a transformation.)

Reduction Strategies for Proving DS Schemes EUF-CMA-secure. Consider an adversary on a DS scheme in the EUF-CMA-experiment above. In such security experiment, the adversary is capable of choosing the forgery message by itself. Hence, in case that the size of the message space $\mathcal{M}$ is superpolynomial (e.g., for $\mathcal{M} = \{0,1\}^k$), there are superpolynomially many possible forgery messages. To argue about security, one usually efficiently

[1] The standard model is an established computational model in cryptography.

reduces a solution to an instance of a computational problem (e.g., factoring) to a successful and efficient adversary on the DS scheme. Hence, the reduction has to "embed" the problem instance into the system's parameters (or into some other "resource" used by the system) and has to hope that the adversary will solve exactly that instance. Since there are superpolynomially many possible forgery messages, "simple guessing" might not be a profitable strategy (since this would lead to a non-efficient reduction, at least in case of a superpolynomially large message space). Further, the reduction should be able to answer signature queries for adversarially chosen messages adaptively. Hence, one carefully has to choose where to embed the problem instance into the system. We note that there are more sophisticated reduction strategies, e.g., partitioning (where, essentially, the reduction is able to partition the message set such that there are messages that can be signed and messages that cannot be signed by the reduction). The partitioning approach yields very efficient schemes in the random oracle model (e.g., [BR96, Cor00, BLS01]) and efficient schemes in the standard model (e.g., [Wat05, HJK11]); unfortunately, some standard-model schemes under the comparably mild Computational Diffie-Hellman (CDH) assumption have relatively large verification keys (e.g., [Wat05]).

Our Contribution. We present a reduction strategy, dubbed confined guessing, to construct efficient DS schemes. This approach was published in [BHJ+13, BHJ+15]. We develop the concept of confined guessing two-staged. First, we define a new cryptographic uilding block named tag-based signatures where the signing and the verification algorithms take a tag as additional input. Further, we define a slightly different and milder form of security for tag-based signatures in comparison to EUF-CMA-security, dubbed m-EUF-naCMA. In an m-EUF-naCMA-security experiment, the adversary has to output message-tag pairs before receiving the verification key. After giving the corresponding signatures and the verification key to the adversary, it eventually outputs a forgery (i.e., a forgery message, a signature, and a forgery tag). The adversary succeeds

- if such a signature (together with the forgery message and the forgery tag) is verified correctly under the verification key,
- the adversary has not queried a signature for the forgery message before, and

– at least one but at most m previously output tag(s) are equal to the forgery tag.

Since the adversary in the m-EUF-naCMA-experiment has to reuse at least one tag (of the polynomially many tags the adversary has output before), we can use this requirement to embed a computational problem associated with the forgery tag into the verification key. Hence, we can guess with significant probability which problem instance the adversary is going to solve. (This is essentially the reason why the concept is called "confined guessing.") The idea is that this milder form of security might be easier achieved and that it would lead to efficient tag-based signature schemes. Secondly, we use a m-EUF-naCMA-secure tag-based DS scheme, a pseudorandom function (PRF), and a chameleon hash (CH) system as building blocks to derive fully secure DS schemes. Essentially, we will use several tag-based signatures in parallel. To this end, we partition the tag space into $\lfloor \log(k) \rfloor$ (disjoint) sets and define for each set a "tag-subset-specific PRF" which can be derived from the "global" PRF. (More details below.) The key generation of the emerged DS scheme samples a PRF key, runs the key generation of the CH system and the key generation of the underlying tag-based scheme to derive a CH function and a trapdoor as well as verification and secret keys, respectively. It then outputs the verification key together with the PRF key, the (descriptions of the) CH function, and the secret key. Signature generation samples a tag for each tag set using the the chameleon hash of the message as input to the corresponding tag-subset-specific PRF. Further, the tag-based signature algorithm is run for each tag together with the secret key and the message. Hence, the signature consists of $\lfloor \log(k) \rfloor$ tag-based signatures. Verification uses the tag-subset-specific PRFs as above and verifies the chameleon hash of the message and the signature using the tag-based verification. If the tag-based verification successfully verifies for all tags under the verification key and the chameleon hash of the message, verification of the DS scheme is successful. For correctness, consider that if the underlying tag-based scheme is correct, the emerged scheme is correct. To prove security, we present an efficient cryptographic transformation from (mild) m-EUF-naCMA- to (full) EUF-CMA-security which will be at the heart of this chapter. Hence, we are able to transform a mildly secure tag-based DS scheme (plus a PRF and a CH system) into a fully secure DS scheme. The verification and secret key sizes are the same as the verification and secret key sizes of the underlying

scheme plus a PRF key and the descriptions of the CH system. The signature size is $\lfloor \log(k) \rfloor$-times as large as the signature size of the tag-based scheme. However, the reduction loss is rather large, but still polynomial in the security parameter. We do the whole transformation modularly in two steps: first, we introduce a slightly weaker security notion in comparison to EUF-CMA, dubbed EUF-dnaCMA-security, and transform any successful and efficient EUF-dnaCMA-adversary on the DS scheme to a successful and efficient m-EUF-naCMA-adversary on the underlying scheme or to a PRF distinguisher. (The EUF-dnaCMA-security experiment can be seen as a non-adaptive version with distinct adversary messages of the EUF-CMA-security experiment.) Secondly, we transform any EUF-dnaCMA-secure DS scheme and a CH system into an EUF-CMA-secure DS scheme. (More technical details follow below.) Further, we mention that a similar strategy in a different context was used by Berman and Haitner [BH12, BH15].

More Technical Details. We start with the first reduction from the DS scheme's EUF-dnaCMA-security to m-EUF-naCMA-security of the underlying tag-based scheme and the PRF indistinguishability property. As already mentioned above, we partition the tag space $\mathcal{T} = \{0,1\}^k$ of the underlying tag-based signature scheme. Concretely, we set $\mathcal{T} = \bigcup_{i=1}^{\lfloor \log k \rfloor} \mathcal{T}_i$, for disjoint subsets $\mathcal{T}_i = \{0,1\}^{\lceil c^i \rceil}$, for "granularity paramater" $c \in \mathbb{N}$, $c > 1$, which is specified globally. The reduction starts by singling out a subset $\mathcal{T}_{i^*}$, for some $i^* \in [\lfloor \log k \rfloor]$, such that

(a) the probability of an $(m+1)$-fold tag collision (for sampling tags uniformly and independently from $\mathcal{T}_{i^*}$) is at most $\varepsilon(k)/2$, where $\varepsilon(k)$ is the advantage of the EUF-dnaCMA-adversary, and
(b) the size of $\mathcal{T}_{i^*}$ is polynomial in the security paramater k, i.e., $|\mathcal{T}_{i^*}| \leq p(k)$, for some suitable polynomial p.

We require (a) to ensure that whenever no $(m+1)$-fold tag collision occurs, the EUF-dnaCMA-adversary has to forge a signature on a message sometimes. Further, we need (b) in a sense that we can later guess with significant probability for which tag in $\mathcal{T}_{i^*}$ the EUF-dnaCMA-adversary forges a signature on a message. (The simulator can then use this forgery as its own for the m-EUF-naCMA-challenger.) To be little bit more concrete, we have a simulator that receives q distinct messages from the EUF-dnaCMA-adversary, samples a PRF key, and uses the "tag-subset-specific" PRFs (which can be derived from the global PRF, details below) with the messages as input to

sample a tag for each subset under each message. Further, the simulator has to make sure that it queries a signature for each tag in $\mathcal{T}_{i^*}$; therefore, it uniformly samples a message from the message space for each (so far unused) tag in $\mathcal{T}_{i^*}$. Then, it outputs all message-tag pair to the m-EUF-naCMA-challenger. (Hence, there are at most $|\mathcal{T}_{i^*}| + \lfloor \log(k) \rfloor \cdot q$ signature queries by the simulator.) This is necessary since the simulator does not know for which forgery message and, thus, for which tag from $\mathcal{T}_{i^*}$, the EUF-dnaCMA-adversary outputs a signature. After receiving all signatures and a verification key from its challenger, the simulator sets up all signatures, the verification, and the PRF key for the EUF-dnaCMA-adversary. Eventually, the EUF-dnaCMA-adversary outputs a forgery message and a signature, and the simulator forwards the message, the tag (derived from the tag-subset-specific PRF), and the i^*-th signature to its own challenger. Hence, whenever the EUF-dnaCMA-adversary forges a signature, then the simulator forges a signature. Further, if no $(m+1)$-fold tag collision for tags in $\mathcal{T}_{i^*}$ (using the PRFs) occurs and the forgery message from the EUF-dnaCMA-adversary is a "fresh" forgery message for the m-EUF-naCMA-challenger, then the simulator is an efficient and successful m-EUF-naCMA-adversary. Additionally, we can construct a PRF distinguisher that uses the simulator described above to quantify the $(m+1)$-fold tag collision. (Note that the simulator above uses the PRF only as a black box.) If $(m+1)$-fold tag collision occurs significantly more often than with uniformly independently sampled tags (see (a)), this yields a contradiction to the PRF's indistinguishability property. (Note that all messages output by the EUF-dnaCMA-adversary are pairwise distinct and, thus, yield to different PRF queries.) Hence, this shows a reduction from the EUF-dnaCMA-security of a DS scheme to the m-EUF-naCMA-security of the underlying tag-based scheme and the indistinguishability property of a PRF.

The second transformation from EUF-dnaCMA-security of a DS scheme and a chameleon hash system to EUF-CMA-security of the emerged DS scheme is very similar to existing transformations, e.g., [KR00]. The difference is that we have to construct a "distinct-message" adversary in the EUF-dnaCMA-security experiment. However, with overwhelming probability (due to the properties of the CH system), we derive distinct "messages" (i.e., chameleon hashes of the EUF-CMA-adversary's messages), for which the simulator wants to see signatures from its EUF-dnaCMA-challenger.

3.1 Preliminaries

EUF-dnaCMA-Security. We say a DS scheme SIG is existentially unforgeable under distinct-message non-adaptive chosen-message attacks (EUF-dnaCMA-secure) if and only if any PPT adversary A has only negligible advantage in the following security experiment. First, A provides messages $(M_i)_{i=1}^q$, for $q = q(k)$, it wants to see signed. In the following, A receives signatures $(\sigma_i)_i := (\mathsf{Sig}(sk, M_i))_i$ and a verification key vk, for $(vk, sk) \leftarrow \mathsf{Gen}(k)$. Eventually, A outputs a signature σ^* on a forgery message M^*. If $\mathsf{Ver}(vk, \sigma^*, M^*) = 1$ and A has only output pairwise distinct messages $((M_i)_{i=1}^q, M^*)$, then the experiment outputs 1. More formally, we define the advantage function for an adversary A as

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

where the experiment $\mathsf{Exp}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG},A}(k)$ is given in Figure 3.1 and SIG is a DS scheme. Then we say SIG is EUF-dnaCMA-secure if and only if for any PPT adversary A the function $\mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG},A}(k)$ is negligible in k.

Experiment $\mathsf{Exp}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG},A}(k)$
$(M_i)_{i=1}^q \leftarrow A(k)$, for $q = q(k)$
$(vk, sk) \leftarrow \mathsf{Gen}(k)$
$(\sigma_i)_{i=1}^q \leftarrow (\mathsf{Sig}(sk, M_i))_{i=1}^q$
$(M^*, \sigma^*) \leftarrow A(vk, (\sigma_i)_{i=1}^q)$
if $\mathsf{Ver}(vk, M^*, \sigma^*) = 1$ and
  for all $i \neq j : M_i \neq M_j \neq M^*$
then return 1 else return 0

Figure 3.1: EUF-dnaCMA-experiment for DS schemes.

Chameleon Hashing. A chameleon hash (CH) scheme CH with message space $\mathcal{M}$, random space $\mathcal{R}$, and hash space $\mathcal{H}$ consists of two algorithms Gen and Col as follows:

Generation. $\mathsf{Gen}(k)$ outputs a hash function and a trapdoor (H, t), for $\mathsf{H} : \mathcal{M} \times \mathcal{R} \to \mathcal{H}$.

Collision. $\mathsf{Col}(t, r, M, M')$, on input trapdoor t, random value $r \in \mathcal{R}$, messages $M, M' \in \mathcal{M}$, outputs a random value r'.

We define correctness and collision resistance of a CH system in the following sense:

Correctness. For all $k \in \mathbb{N}$, for all $(H, t) \leftarrow \mathsf{Gen}(k)$, for all $r \leftarrow \mathcal{R}$, for all $M, M' \in \mathcal{M}$, for all $r' \leftarrow \mathsf{Col}(t, r, M, M')$, we have $\mathsf{H}(M, r) = \mathsf{H}(M', r')$.

Collision Resistance. For any PPT algorithm D, the function

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

such that $\mathsf{H}(M, r) = \mathsf{H}(M', r')$ with $(M, r) \neq (M', r')$, is negligible in k, for $(\mathsf{H}, t) \leftarrow \mathsf{Gen}(k)$. Further, given only H and M, the distribution of r is uniform. (Note that t is not given to D.)

3.2 (Mildly-Secure) Tag-Based Signatures

Tag-Based (Digital) Signatures. A tag-based signature scheme TSIG with message space $\mathcal{M}$ and tag space $\mathcal{T}$ consists of three PPT algorithms $(\mathsf{TGen}, \mathsf{TSig}, \mathsf{TVer})$ as follows:

Key generation. $\mathsf{TGen}(k)$, on input security parameter k, outputs public parameter pp, and verification and secret keys (pp, vk, sk). (Analogously to DS schemes, we assume that Sig and Ver have implicitly access to pp.)

Sign. $\mathsf{TSig}(sk, M, t)$, on input sk, message $M \in \mathcal{M}$, and tag $t \in \mathcal{T}$, outputs a signature σ.

Verification. $\mathsf{TVer}(vk, \sigma, M, t)$, on input vk, σ, message M, and tag t, outputs $b \in \{0, 1\}$.

We define correctness and (a mild form of) security for tag-based signatures:

Correctness. For all $k \in \mathbb{N}$, for all $(vk, sk) \leftarrow \mathsf{TGen}(k)$, for all $M \in \mathcal{M}$, for all $t \in \mathcal{T}$, for all $\sigma \leftarrow \mathsf{Ext}(sk, M, t)$, we have $\mathsf{Ver}(vk, \sigma, M, t) = 1$.

m-EUF-naCMA-security. For $m \in \mathbb{N}$, we say that a tag-based signature scheme TSIG is existentially unforgeable under distinct-message(s) non-adaptive chosen-message attacks with at least one but at most m forgery-tag collisions (m-EUF-naCMA-secure) if and only if any PPT adversary A has only negligible advantage in the following security experiment. First, A (non-adaptively) provides at most $q = q(k)$ message-tag pairs $(M_i, t_i)_{i=1}^q$. Further, the experiment computes signatures $(\sigma_i)_{i=1}^q := (\mathsf{Sig}(sk, M_i, t_i))_{i=1}^q$ and sends $(vk, (\sigma_i)_{i=1}^q)$ to A, for honestly generated $(vk, sk) \leftarrow \mathsf{TGen}(k)$. Eventually, A outputs (M^*, σ^*, t^*). We say that A is valid if

- A has output a message $M^* \notin \{M_i\}_{i=1}^q$ and
- at least one tag but at most m tags in $(t_i)_{i=1}^q$ are equal to t^*.

If $\mathsf{Ver}(vk, \sigma^*, M^*, t^*) = 1$ and A is valid, then the experiment outputs 1. More formally, we define the advantage function for an adversary A as

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

where the experiment $\mathsf{Exp}^{m\text{-euf-nacma}}_{\mathsf{TSIG},A}(k)$ is given in Figure 3.2 and TSIG is a tag-based signature scheme. Then we say TSIG is m-EUF-naCMA-secure if and only if the function $\mathsf{Adv}^{m\text{-euf-nacma}}_{\mathsf{TSIG},A}(k)$ is negligible in k, for any PPT adversary A.

Experiment $\mathsf{Exp}^{m\text{-euf-nacma}}_{\mathsf{TSIG},A}(k)$
$(M_i, t_i)_{i=1}^q \leftarrow A(k)$, for $q = q(k)$
$(vk, sk) \leftarrow \mathsf{TGen}(k)$
$(\sigma_i)_{i=1}^q \leftarrow (\mathsf{TSig}(sk, M_i, t_i))_{i=1}^q$
$(M^*, \sigma^*, t^*) \leftarrow A(vk, (\sigma_i)_{i=1}^q)$
if $\mathsf{TVer}(vk, M^*, \sigma^*, t^*) = 1$ and A is valid
then return 1 else return 0

Figure 3.2: The m-EUF-naCMA experiment for tag-based signature schemes.

3.3 From Mild to Distinct-Message Non-Adaptive Security

We construct non-adaptively secure digital signature schemes from mildly secure tag-based signature schemes and a PRF. (We will construct fully secure, i.e., EUF-CMA-secure, digital signature schemes in the next section.) Let $\mathsf{TSIG} = (\mathsf{TGen}, \mathsf{TSig}, \mathsf{TVer})$ with message space $\mathcal{M}$ and tag space $\mathcal{T} = \{0,1\}^k$ be a tag-based signature scheme and let PRF be a PRF with key space $\mathcal{K}$, domain space $\mathcal{M}$, and range $\mathcal{T}$. We partition the tag space such that $\mathcal{T} = \bigcup_{i=1}^{\mu} \mathcal{T}_i$, for $\mu := \lfloor \log_c(k) \rfloor$, (pairwise disjoint) subsets $(\mathcal{T}_i)_i$, and $|\mathcal{T}_i| = 2^{\lceil c^i \rceil}$, for granularity parameter $c \in \mathbb{N}$, with $c > 1$. We construct a signature scheme $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ with message space $\mathcal{M}$ as follows:

Key generation. $\mathsf{Gen}(k)$ outputs $(pp, vk, sk) := (K, vk', sk')$, for $K \leftarrow \mathcal{K}$, and for $(vk', sk') \leftarrow \mathsf{TGen}(k)$. (Note that pp is implicitly given to Sig and Ver.)

Sign. $\mathsf{Sig}(sk, M)$, on input sk and message $M \in \mathcal{M}$, computes a tag $t_i := \mathsf{PRF}_K^{\mathcal{T}_i}(M)$ and a tag-based signature $\sigma_i := \mathsf{TSig}(sk, M, t_i)$, for all $i \in [\mu]$, and outputs the signature $\sigma := (\sigma_i)_{i=1}^{\mu}$.

Verification. $\mathsf{Ver}(vk, \sigma, M)$, on input vk, signature $\sigma =: (\sigma_i)_{i=1}^{\mu}$ and message M, outputs $\bigwedge_{i=1}^{\mu} \mathsf{TVer}(vk, \sigma_i, M, t_i)$, for $(t_i)_{i=1}^{\mu} := (\mathsf{PRF}_K^{\mathcal{T}_i}(M))_{i=1}^{\mu}$.

(See also Figure 3.3.) For correctness note that if TSIG is correct, then SIG is correct.

Theorem 3.3.1. *If* PRF *is a PRF as above and* $\mathsf{TSIG} = (\mathsf{TGen}, \mathsf{TSig}, \mathsf{TVer})$ *with tag space* $\mathcal{T} = \bigcup_{i=1}^{\mu} \mathcal{T}_i$ *(as above, i.e., for* $\mu = \lfloor \log_c(k) \rfloor$*) and message space* $\mathcal{M}$*, where* $|\mathcal{M}| > m \cdot |\mathcal{T}_{i^*}|$*, for* i^* *as in Lemma 3.3.3, is an* m*-EUF-naCMA-secure tag-based signature scheme, then* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ *with message space* $\mathcal{M}$ *is an EUF-dnaCMA-secure signature scheme. Concretely, for any successful PPT adversary* A *with advantage function* $\varepsilon(k) := \mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG},A}(k) > 1/p'(k)$*, for some polynomial* p' *and infinitely many* k*, and number of signature queries* q *on* SIG*, there are PPT adversaries* A_1 *with number of signature queries at most* $2 \cdot (2q^{m+1}/\varepsilon(k))^{c/m} + \mu q$ *on* TSIG *and* A_2 *on* PRF*, and a polynomial* $p(k)$ *such that*

$$\mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG},A}(k)/2 \leq \mathsf{Adv}^{m\text{-}\mathsf{euf\text{-}nacma}}_{\mathsf{TSIG},A_1}(k) + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF},A_2}(k) + p(k)/|\mathcal{M}|. \quad (3.1)$$

Algorithm $\mathsf{Gen}(k)$
$(vk, sk) \leftarrow \mathsf{TGen}(k)$
$K \leftarrow \mathcal{K}$
return (K, H, vk, sk)

Algorithm $\mathsf{Sig}(sk, M)$
$(t_i)_{i=1}^{\mu} := (\mathsf{PRF}_K^{\mathcal{T}_i}(M))_i$
$(\sigma_i)_i := (\mathsf{TSig}(sk, M, t_i))_i$
return $(\sigma_i)_i$

Algorithm $\mathsf{Ver}(vk, (\sigma_i)_i, M)$
$(t_i)_i := (\mathsf{PRF}_K^{\mathcal{T}_i}(M))_i$
return $\bigwedge_i \mathsf{TVer}(vk, \sigma_i, M, t_i)$

Figure 3.3: An EUF-dnaCMA-secure signature scheme.

Proof. We first state two lemmas which are helpful in the reduction below.

Lemma 3.3.2 (Restated from [HJK11, Lemma 2.3])**.** *For a finite set $\mathcal{T}$, we have that for $(t_i)_{i=1}^{q} \leftarrow (\mathcal{T})^q$ the probability that there exist pairwise distinct $(i_j)_{j=1}^{m+1} \in ([q])^{m+1}$ such that $t_{i_1} = \cdots = t_{i_{m+1}}$ is at most $q^{m+1}/|\mathcal{T}|^m$.*

Lemma 3.3.3. *For $i^* := \lceil \log_c \left(\frac{1}{m} \cdot \log_2 \left(2 \cdot q^{m+1}/\varepsilon(k) \right) \right) \rceil$, we have that*

$$\Pr\left[\exists \text{ pairwise distinct } (i_j)_{j=1}^{m+1} \in [q] \text{ such that } t_{i^*,i_1} = \cdots = t_{i^*,i_{m+1}}\right] \leq \frac{\varepsilon(k)}{2}, \tag{3.2}$$

with probability over $(t_{i^*,j})_{j=1}^{q} \leftarrow (\mathcal{T}_{i^*})^q$, and $|\mathcal{T}_{i^*}| \leq 2 \cdot (2q^{m+1} \cdot p'(k))^{c/m}$. (For all (other) variables defined as above.)

Proof. Applying Lemma 3.3.2 yields

$$\Pr\left[\exists \text{ pairwise distinct } (i_j)_{j=1}^{m+1} \in [q] \text{ such that } t_{i^*,i_1} = \cdots = t_{i^*,i_{m+1}}\right] \leq \frac{q^{m+1}}{|\mathcal{T}_{i^*}|^m},$$

and $q^{m+1}/|\mathcal{T}_{i^*}|^m = q^{m+1}/2^{\lceil c^{i^*} \rceil \cdot m} \leq q^{m+1}/2^{c^{i^*} \cdot m} \leq q^{m+1} \cdot \varepsilon(k)/2q^{m+1} = \varepsilon(k)/2$ which shows Equation 3.2. Further, $|\mathcal{T}_{i^*}| = 2^{\lceil c^{i^*} \rceil} \leq 2 \cdot 2^{c^{i^*+1}} =$

$2 \cdot (2q^{m+1}/\varepsilon(k))^{c/m} \leq 2 \cdot (2q^{m+1} \cdot p'(k))^{c/m}$. □

We proceed with the reduction; i.e., we describe and analyze A_1 and A_2 as follows.

Description. A_1 receives (distinct) messages $(M_i)_{i=1}^q$ from A, samples a PRF-seed $K \leftarrow \mathcal{K}$ (given black-box access to PRF), and lists message-tag pairs $\mathcal{L} := (M_i, t_{i^*,i})_{i=1}^q$, for $(t_{i^*,i})_i := (\mathsf{PRF}_K^{\mathcal{T}_{i^*}}(M_i))_i$, for

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

(See Lemma 3.3.3 for details. Further, we assume that A_1 knows $\varepsilon(k)$.) If an $(m+1)$-fold tag collision for tags $(t_{i^*,i})_i$ occurs, A_1 aborts. For all $(j,i) \in [\mu] \times [q]$, with $j \neq i^*$, A_1 computes $t_{j,i} := \mathsf{PRF}_K^{\mathcal{T}_j}(M_i)$ and sets $\mathcal{L} := (\mathcal{L}, (M_i, t_{j,i})_{j,i})$. Further, for each (so far unused) tag $t'_{i^*,i'} \in \mathcal{T}_{i^*} \setminus \{t_{i^*,i}\}_i$, A_1 picks a (dummy) message $M'_{i^*,i'} \leftarrow \mathcal{M}$ and sets $\mathcal{L} := (\mathcal{L}, (M'_{i^*,i'}, t'_{i^*,i'})_{i'})$. (Hence, A_1 lists at least one message-tag pair for each tag in $\mathcal{T}_{i^*}$.) Finally, A_1 outputs $\mathcal{L}$ and, in turn, receives a verification key vk, signatures $(\sigma_{i,j})_{i,j}$, for $(i,j) \in [\mu] \times [q]$, and (dummy) signatures $(\sigma'_{i^*,i'})_{i'}$ for message-tag pairs $((M'_{i^*,i'}), t'_{i^*,i'})_{i'}$ from its challenger. A_1 is now ready to provide (K, vk) and signatures $(\sigma_{i,j})_{i,j}$ for A. Eventually, A outputs a message M^* and a signature $\sigma^* = (\sigma_i^*)_{i=1}^{\mu}$. If $M^* \in \{M'_{i^*,i'}\}_{i'}$, then A_1 aborts. Otherwise, A_1 outputs $(M^*, \sigma_{i^*}^*, t^*)$, for $t^* := \mathsf{PRF}_K^{\mathcal{T}_{i^*}}(M^*)$, to its challenger. (In Figure 3.4, we give a schematic representation of A_1.)

A_2 has oracle access either to a pseudorandom function or to a truly random function. (Note that A_1 uses PRF as a black-box.) Now, A_2 simulates A_1. If the probability for $(m+1)$-fold tag collisions for tags $(t_{i^*,i})_{i=1}^q$ as computed above is significantly larger than $\varepsilon(k)/2$, then A_2 outputs 0, otherwise 1. (Note that A only outputs distinct messages $(M_i)_{i=1}^q$. Hence, the input to $\mathsf{PRF}_K^{\mathcal{T}_{i^*}}$ is different for each query. Further, we assume that A_2 knows $\varepsilon(k)$.)

Analysis. Let abort be the event that A_1 aborts. Note that (K, vk) and all signatures $(\sigma_{i,j})_{i,j}$ for A are constructed correctly in the sense of SIG. Now, whenever A forges a signature σ^* on a forgery message M^* under vk (i.e., $\mathsf{Ver}(vk, M^*, \sigma^*) = 1$ and $M^* \notin \{M_i\}_i$), it must hold that $\mathsf{TVer}(vk, M^*, \sigma_{i^*}^*, t^*) = 1$, for $t^* := \mathsf{PRF}_K^{\mathcal{T}_{i^*}}(M^*)$ and i^* as chosen above. Hence, A_1 forges a signature, whenever A does. Further, A_1 has to be a valid

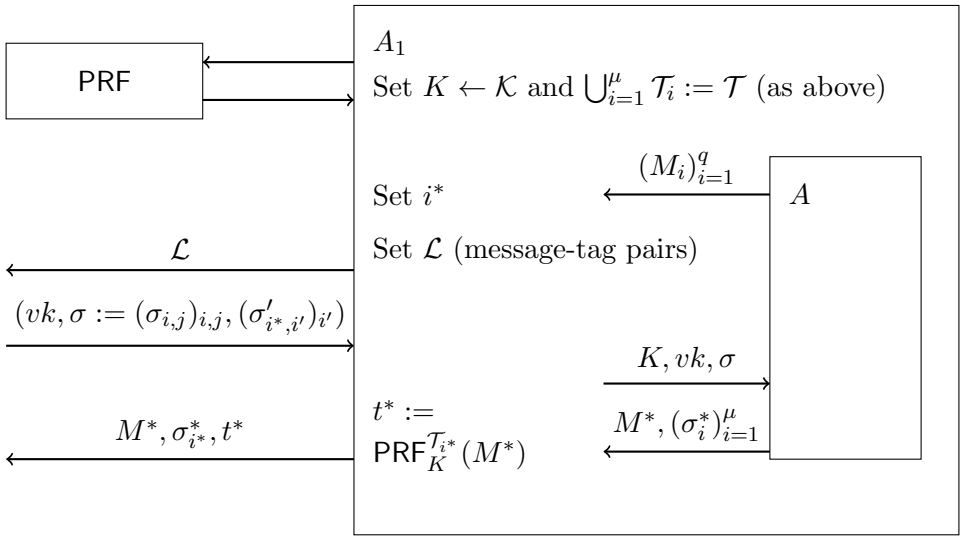

Figure 3.4: Schematic representation of (non-aborting) A_1.

adversary in the sense of m-EUF-naCMA (i.e., (a) A_1 has output a message $M^* \notin \{M_i'\}_{i=1}^{|\mathcal{L}|}$, for all messages M_i' contained in list $\mathcal{L}$; and (b) at least one tag but at most m tags are equal to t^*). Hence, A_1 is valid, whenever abort does not occur. Concerning (a), the probability that $M^* \in \{M_i'\}_{i=1}^{|\mathcal{L}|}$ is at most $m|\mathcal{T}_{i^*}|/|\mathcal{M}|$. (Note that $|\mathcal{T}_{i^*}|$ is polynomial in k due to Lemma 3.3.3 and the dummy messages $\{M'_{i^*,i'}\}_{i'}$ are independent of A's view.) For (b), since A_1 has output a message-tag pair for each tag in $\mathcal{T}_{i^*}$, the forgery tag t_{i^*} must be a reused tag. Further, by Lemma 3.3.3, the probability that $(m+1)$-fold tag collisions for uniformly and independently sampled tags from $\mathcal{T}_{i^*}$ occur is bounded by at most $\varepsilon(k)/2$. Note that the tags in the system are chosen using $\mathsf{PRF}_K^{\mathcal{T}_{i^*}}$. Now, if the probability of an $(m+1)$-fold tag collision with $\mathsf{PRF}_K^{\mathcal{T}_{i^*}}$ is larger than $\varepsilon(k)/2$, then this yields a PRF distinguisher A_2 (as described above). Thus,

$$\Pr[\mathsf{abort}] \leq \varepsilon(k)/2 + \mathsf{Adv}^{\mathsf{prf}}_{\mathsf{PRF},A_2}(k) + m|\mathcal{T}_{i^*}|/|\mathcal{M}|,$$

for $\varepsilon(k) = \mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG},A}(k)$ and

$$|\,\mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG},A}(k) - \mathsf{Adv}^{m\text{-}\mathsf{euf\text{-}nacma}}_{\mathsf{TSIG},A_1}(k)\,| \leq \Pr[\mathsf{abort}],$$

shows Equation 3.1. □

3.4 From Distinct-Message Non-Adaptive to Full Security

In this section, we construct fully secure DS schemes from distinct-message non-adaptively secure DS schemes. (We use chameleon hashing in a similar way as it is used in standard constructions to get fully secure signatures from non-adaptively secure signatures; e.g., see [KR00] or [HW09b, Lemma 2.3].)

Let $\mathsf{SIG}' = (\mathsf{SIG}'.\mathsf{Gen}, \mathsf{SIG}'.\mathsf{Sig}, \mathsf{SIG}'.\mathsf{Ver})$ with message space $\mathcal{M}$ be a signature scheme and let $\mathsf{CH} = (\mathsf{CH}.\mathsf{Gen}, \mathsf{CH}.\mathsf{Col})$ be a chameleon hash system with message space $\mathcal{M}$, random space $\mathcal{R}$, and hash space $\mathcal{H}$. We construct a DS scheme $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ with message space $\mathcal{M}$ as follows:

Key generation. $\mathsf{Gen}(k)$ outputs $(pp, vk, sk) := ((pp', \mathsf{H}), vk', sk')$, for H as first output of $\mathsf{CH}.\mathsf{Gen}(k)$, and for $(pp, vk', sk') \leftarrow \mathsf{SIG}'.\mathsf{Gen}(k)$. (Note that pp is implicitly given to Sig and Ver.)

Sign. $\mathsf{Sig}(sk, M)$, on input sk and message $M \in \mathcal{M}$, samples $r \leftarrow \mathcal{R}$, computes $\sigma := \mathsf{SIG}'.\mathsf{Sig}(sk, \mathsf{H}(M, r))$, and outputs (σ, r).

Verification. $\mathsf{Ver}(vk, (\sigma, r), M)$, on input vk, "signature" (σ, r), and message M, outputs $\mathsf{SIG}'.\mathsf{Ver}(vk, \sigma, \mathsf{H}(M, r))$.

(See also Figure 3.5.) For correctness note that if SIG' is correct, then SIG is correct.

Theorem 3.4.1. *If* CH *is a chameleon hash system as above and* $\mathsf{SIG}' = (\mathsf{SIG}'.\mathsf{Gen}, \mathsf{SIG}'.\mathsf{Sig}, \mathsf{SIG}'.\mathsf{Ver})$ *with message space* $\mathcal{M}$ *is an EUF-dnaCMA-secure digital signature scheme, then* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ *with message space* $\mathcal{M}$ *is an EUF-CMA-secure signature scheme. Concretely, for a successful PPT EUF-CMA-adversary* A *with number of signature queries* $q = q(k)$ *on* SIG*, there is an successful PPT EUF-dnaCMA-adversaries* A_1 *with number of signature queries* q' *with* $q \leq q' \leq q'(k)$*, for some suitable polynomial* $q'(k)$*, on* SIG' *or a chameleon hash distinguisher* A_2 *on* CH *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},A}(k) \leq\ & \mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG}',A_1}(k) + \mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},A_2}(k) \\ & + \mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},D}(k) + (q')^2/(|\mathcal{R}| \cdot |\mathcal{M}|),\end{aligned} \tag{3.3}$$

for any PPT chameleon hash distinguisher D*.*

Algorithm $\mathsf{Gen}(k)$
$(pp, vk, sk) \leftarrow \mathsf{SIG}'.\mathsf{Gen}(k)$
$(\mathsf{H}, t) \leftarrow \mathsf{CH}.\mathsf{Gen}(k)$
return $((pp, \mathsf{H}), vk, sk)$

Algorithm $\mathsf{Sig}(sk, M)$
$r \leftarrow \mathcal{R}$
$\sigma := \mathsf{SIG}'.\mathsf{Sig}(sk, \mathsf{H}(M, r))$
return (σ, r)

Algorithm $\mathsf{Ver}(vk, (\sigma, r), M)$
return $\mathsf{SIG}'.\mathsf{Ver}(vk, \sigma, \mathsf{H}(M, r))$

Figure 3.5: An EUF-CMA-secure signature scheme.

Proof. **Description.** A_1 computes $(\mathsf{H}, t) \leftarrow \mathsf{CH}.\mathsf{Gen}(k)$. Further, A_1 samples pairs $(M_i, r_i)_{i=1}^{q'} \leftarrow (\mathcal{M} \times \mathcal{R})^{q'}$ and outputs $(\mathsf{H}(M_i, r_i))_i$ to its EUF-dnaCMA-challenger. If $|\{\mathsf{H}(M_i, r_i)\}_i| < q'$, then A_1 aborts. Otherwise, A_1 receives public parameters, the verification key, and signatures $(pp, vk, (\sigma_i)_i)$ from its challenger and starts A with $((pp, \mathsf{H}), vk)$. During the reduction, A might query signatures and A_1 answers them as follows. On query $M_{i'}$, A_1 computes $r'_{i'} \leftarrow \mathsf{CH}.\mathsf{Col}(t, r_{i'}, M_{i'}, M'_{i'})$, where $i' \in [q']$ is the index of that query, and returns signature-randomness pair $(\sigma_{i'}, r'_{i'})$. Eventually, A outputs $(M^*, (\sigma^*, r^*))$. If $\mathsf{H}(M^*, r^*) = \mathsf{H}(M_i, r_i)$, for some $i \in [q]$, then A_1 aborts. Otherwise, A_1 forwards $(\mathsf{H}(M^*, r^*), \sigma^*)$ to its own challenger. (In Figure 3.6, we give a schematic representation of A_1.)

A_2 receives some chameleon hash function H from its challenger and runs $(pp, vk, sk) \leftarrow \mathsf{SIG}'.\mathsf{Gen}(k)$. Further, A_2 gives $((pp, \mathsf{H}), vk)$ to A and answers signature queries as $(\sigma_i, r_i) := \mathsf{Sig}(sk, M_i)$ (using H as chameleon hash function within Sig), for some query $M_i \in \mathcal{M}$, for all $i \in [q]$. Eventually, A outputs $(M^*, (\sigma^*, r^*))$. If $\mathsf{H}(M^*, r^*) \neq \mathsf{H}(M_i, r_i)$, for all $i \in [q]$, then A_2 aborts. Otherwise, A_2 outputs $(M_{i'}, r_{i'}, M^*, r^*)$, for some $i' \in [q]$ with $\mathsf{H}(M^*, r^*) = \mathsf{H}(M_{i'}, r_{i'})$, to its own challenger. (In Figure 3.7, we give a schematic representation of A_2.)

Analysis. Note that all $((pp, \mathsf{H}), vk)$ and all signatures $(\sigma_i)_i$ for A are cor-

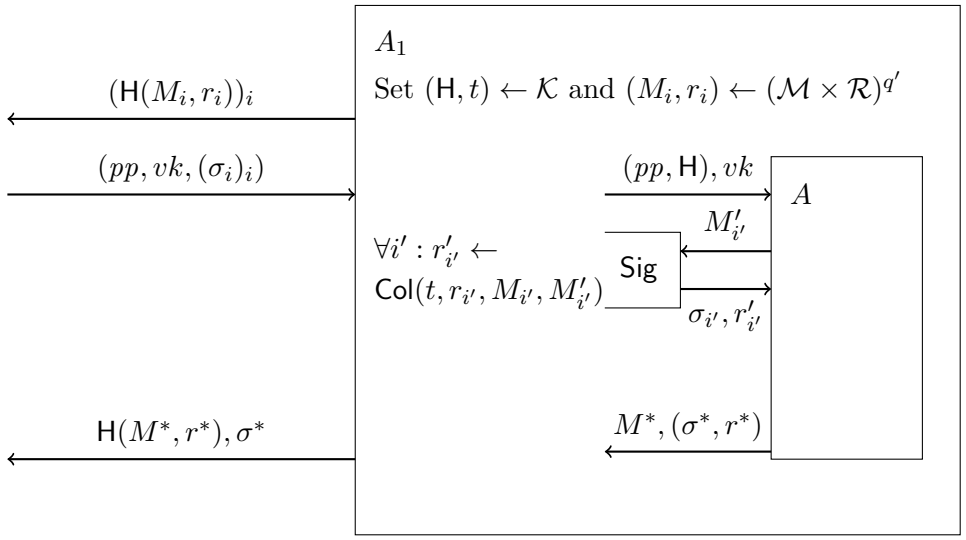

Figure 3.6: Schematic representation of (non-aborting) A_1.

rectly distributed in the sense of SIG. Let col be the event that $\mathsf{H}(M^*, r^*) = \mathsf{H}(M_{i'}, r_{i'})$, for some $i' \in [q]$ and (M^*, r^*) from the A-forgery. We have that $\Pr[\mathsf{col}] \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},A_2}(k)$, for a PPT-forger A and A_2 as above. Further, let abort_{A_1} be the event that A_1 aborts. We have that $\Pr[\mathsf{abort}_{A_1}] \leq \Pr[\mathsf{col}] + \mathsf{Adv}^{\mathsf{cr}}_{\mathsf{CH},D}(k) + (q')^2/(|\mathcal{M}| \cdot |\mathcal{R}|)$, for any efficient CH distinguisher D. (We use Lemma 3.3.2 to give an upper bound for 1-fold collisions while sampling q' times from $\mathcal{M} \times \mathcal{R}$. Further, if no 1-fold collisions occurs, there might still be CH collisions which are bounded by collision resistance property of the CH system. Note that the message-randomness pairs are independent of A's view.) Further, it is clear that whenever an efficient and successful A forges a signature and A_1 does not abort, then A_1 is an efficient and successful EUF-dnaCMA-adversary. Thus,

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{SIG},A}(k) \leq \mathsf{Adv}^{\mathsf{euf\text{-}dnacma}}_{\mathsf{SIG}',A_1}(k) + \Pr[\mathsf{abort}_{A_1}]$$

shows Equation 3.3. □

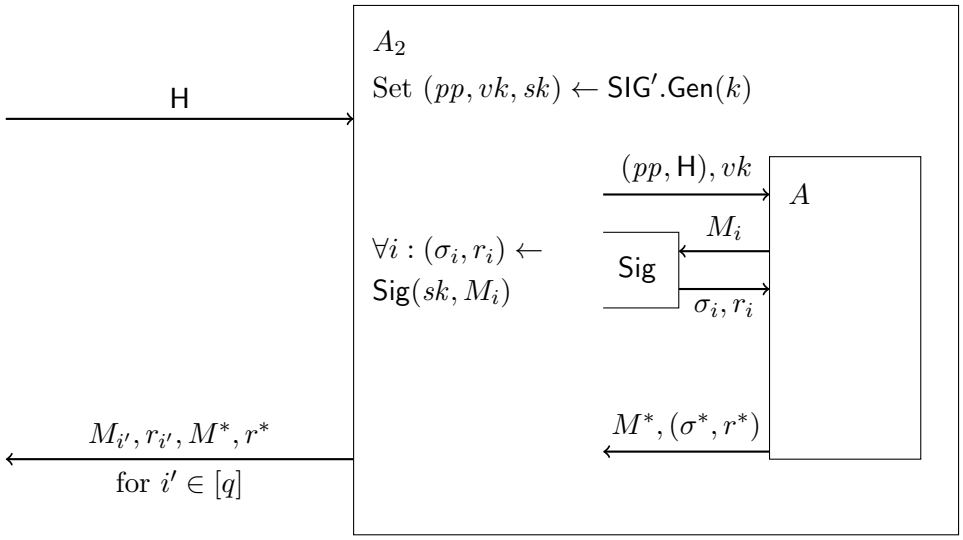

Figure 3.7: Schematic representation of (non-aborting) A_2.

3.5 With a View to Fully Secure Signature Instantiations

The confined guessing concept gives rise to very efficient fully secure digital signature instantiations. This is shown in [BHJ+13, BHJ+15] and we explicitly mention that these instantiations are not part of this thesis and we only describe their efficiency here (in terms of the size of the parameters and computation time).

In [BHJ+13, BHJ+15], we give m-EUF-naCMA-secure tag-based digital signature instantiations under the computational Diffie-Hellman (CDH), the RSA, and the Short Integer Solutions (SIS) assumptions. Applying the confined guessing technique from Section 3.3 and the transformation from Section 3.4, we are able to directly and modularly obtain EUF-CMA-secure digital signatures from the CDH, RSA, and SIS assumptions.

For CDH and RSA, we further optimized the system's parameters (verification key vk, secret key sk, and signature σ) in [BHJ+13, BHJ+15]. This resulted in the first fully secure CDH-based signature scheme with such compact verification keys and in more efficient (in terms of computation) fully secure RSA-based schemes. (Both in comparison to existing CDH- or RSA-based schemes, respectively, at that time.) For SIS, we presented an al-

ternative to existing solutions. See Table 3.1 for a brief overview with the respective parameters. However, we stress that our security reduction has a rather large polynomially overhead.

Assumption	$\lvert vk\rvert$	$\lvert sk\rvert$	$\lvert\sigma\rvert$
CDH	$\mathbf{O}(\log(k))$	$\mathbf{O}(1)$	$\mathbf{O}(1)$
RSA	$\mathbf{O}(1)$	$\mathbf{O}(1)$	$\mathbf{O}(1)$
SIS	$\mathbf{O}(n \cdot n')$	$\mathbf{O}(n')$	$\mathbf{O}(\log(k) \cdot n')$

Table 3.1: For CDH and RSA, vk- and σ-quantities are given in number of group elements. sk-quantities are given in number of $\mathbb{Z}_p$-elements, for $p = p(k)$. For SIS, quantities are given in number of $\mathbb{Z}_p$-elements, where n, n' are the SIS matrix dimensions.

Chapter 4

(Almost) Tight IBE Security

Identity-Based Encryption. Loosly speaking, an identity-based encryption (IBE) scheme is a cryptographic building block that is capable of encrypting a message under a public identity (e.g., a bitstring or e-mail address) and some master public key and public parameters. Further, a master secret key allows to generate user secret keys for identities. These user secret keys can then be used for decryption. (See below for a more detailed definition.) The idea of IBE originates from [Sha84] and (first) instantiation candidates were given in [DQ87, Tan88, TI89, MY91, MY93, LL93, MY96, HJW00]. (Unfortunately, some of the these candidates suffer from certain restrictions, e.g., rely on tamper-resistant hardware, do not allow (many) users to collude, or need heavy computations.)

More Constructions of IBE Schemes. The Boneh-Franklin IBE [BF01] is considered to be the first efficient IBE construction and is based on a Diffie-Hellman-related problem in pairing-friendly groups in the random oracle model[1]. (The work of [BF01, BF03] also presented a formal model of "full" IBE security.) A non-pairing-based IBE scheme based on the difficulty of computing quadratic residues modulo a composite was proposed in [Coc01] (also in the random oracle model). More constructions in the random oracle model are, e.g., [HL02, GS02, GPV08]. The first standard-model secure IBE systems were given in [CHK03, CHK04, BB04a], which

[1]The random oracle model is an idealized security model in cryptography. See [BR93] for an introduction and details.

build on the work of [GS02], but offer only a weaker form of security compared to full IBE security, dubbed "selective-ID" security [CHK03]. Another standard-model secure system was given in [HK04]; however, the proposed IBE is only k-resilient, i.e., the system does not allow that more than k users to collude. Full IBE security in the standard model was given by [BB04b] (for an impractical scheme) and [Wat05] (for a practical scheme). ([BB04a] notes that one can generically transform any selective-ID secure IBE to a fully secure IBE. Unfortunately, the transformation is superpolynomial in the security parameter, at least in case of a superpolynomial identity space.) All standard-model-secure schemes mentioned above are based on pairings and more pairings-related constructions of (selective-ID or fully secure) IBEs in standard model with various efficiency and tightness guarantees can be found in, e.g., [Gen06, BR09, Wat09, LW10, Lew12, CLL+13, CW13, JR13, CLL+14, BKP14, Wee14, HKS15]. Other recent standard-model secure IBE schemes are based on lattices, e.g., [AB09, ABB10a, ABB10b, CHKP10].

Security Notion(s) for IBE Schemes. IBE indistinguishability under chosen-plaintext attacks (IBE-IND-CPA) [BF03] is considered to be a standard security notion for IBE schemes. In IBE-IND-CPA, the adversary gets the honestly generated public parameters and master public key. During the experiment, the adversary may query secret keys for identities of its choice. At some point, the adversary outputs a challenge identity together with two equal-length challenge messages. The experiment samples a bit uniformly at random and sends one of the two messages encrypted (depending on the bit) under the challenge identity to the adversary. Eventually, the adversary outputs a guess which challenge message was encrypted and succeeds if the guess is correct and it has never queried a user secret key for the challenge identity. We say that an IBE scheme is IBE-IND-CPA-secure if any PPT adversary succeeds in the previous experiment only with probability negligibly larger that 1/2. (We also refer to this notion as full IBE security. Note that this security notion deals only with one instance and one challenge ciphertext.)

Goals in Constructing IBE Schemes. It is appealing to construct (at least) IBE-IND-CPA-secure schemes with "short" public and secret parameters, master public and master secret keys, user secret keys, and ciphertexts under simple assumptions in the standard model with efficient parameter generation, master keys generation, user secret key generation, encryption, and decryption. In this sense, a very efficient IBE with constant-size pub-

lic and secret parameters, master public and master secret keys, user secret keys, and ciphertexts under the decisional linear (DLIN) assumption is given in [Wat09]. Further, it is preferable to give a tight or at least an almost tight security reduction since this usually leads to more efficient instantiations. (See paragraph below for an explanation.) Loosely speaking, a tightly secure reduction in the IBE-IND-CPA-sense does not depend on the number of user secret key queries. The first (almost) tightly IBE-IND-CPA-secure IBE scheme in the standard model under the d-LIN assumption was given in [CW13]. Unfortunately, the master public key and the master secret key sizes in the instantiations are linear in the security parameter, all other parameters, e.g., user secret keys and ciphertexts, are constant-size.

More on (Almost) Tight IBE-IND-CPA-Security Reductions from Concrete IBE Schemes. To argue about concrete IBE-IND-CPA-security of a concrete IBE scheme, one efficiently reduces a PPT problem-instance solver of the underlying computational problem to any PPT and successful adversary on the IBE scheme. Hence, the reduction tries to "use" the (significant) success probability ε of a potentially successful IBE-adversary to solve a given instance of the underlying problem with (significant) probability $\varepsilon' \geq \varepsilon/L$, for some "loss" L. If $L \in \mathbf{O}(1)$, we say a reduction is tight and if $L \in \mathbf{O}(k)$, then we call a reduction almost tight. Following [BDJR97] and related work (see below), we treat all adversary (resource) queries separately. A consequence in the IBE-IND-CPA-setting is that an (almost) tight reduction can not depend on the number of user secret key queries. Such (almost) tight security reduction ensures that any efficient IBE adversary, which is successful with probability ε, yields an efficient algorithm that has roughly the same running time and success probability $\varepsilon' \geq \varepsilon/L$, for $L \in \mathbf{O}(k)$, on solving a given problem instance. In a practical setting, this often translates to the usage of shorter keys and, hence, would usually yield more efficient IBE instantiations.

Tight Security in General and the Multi-Instance, Multi-Ciphertext Setting. In a more general context, tight security guarantees were considered at least in, e.g., [BKR94, BGR95, BCK96, BDJR97] in the private-key encryption setting. The works of, e.g., [BBM00, HJ12, LJYP14] deal with tightness guarantees in the public-key encryption (PKE) setting. In the IBE setting, the schemes of [Gen06, GH09] and of [CW13, BKP14, HKS15] are provided with tight security reductions in the standard model under non-

simple and simple assumptions, respectively. Even more generally, tightness in the "multi-instance, multi-ciphertext" PKE setting is considered in, e.g., [BBM00, HJ12, LJYP14]. The multi-instance, multi-ciphertext setting captures a more realistic scenario, in which there are many PKE instances (and, hence, many users) with many ciphertexts per instance. [BBM00] shows that an IND-CPA-secure[2] PKE scheme is generically secure in the multi-instance, multi-ciphertext setting. Unfortunately, the security guarantees degrade in the number instances and challenge ciphertexts per instance, i.e., we have that $L \in \mathbf{O}(\mu q k)$, for "loss" L, for the number of instances μ, and for the number of challenge ciphertexts per instance q. Such (generic) reduction is considered non-tight, since L is not constant in k. However, more specific standard-model PKE schemes are tightly IND-CPA-secure in the multi-instance, multi-ciphertext setting, i.e., the El Gamal scheme under the decisional Diffie-Hellman (DDH) assumption [BBM00]. Further, the first tightly standard-model IND-CCA-secure[3] PKE scheme under a standard assumption in the multi-instance, multi-ciphertext setting was proposed in [HJ12]. Concerning IBE, the schemes of [CW13, BKP14] do not consider the multi-instance, multi-ciphertext setting.

Our Contribution. In the following, we will construct a standard-model (almost) tightly secure IBE scheme in the multi-instance, multi-ciphertext setting via a cryptographic transformation. The approach was published in [HKS15] and constitutes the heart of this chapter. We first give a natural extension of the IBE-IND-CPA-security notion which consists of multiple instances and ciphertexts, dubbed (μ, q)-IBE-IND-CPA. In an (μ, q)-IBE-IND-CPA-experiment, the adversary receives honestly generated master public keys of all μ instances. During the experiment, it may adaptively query user secret keys and up to q challenge ciphertexts per instance. The challenge ciphertexts are created under a challenge identity, the corresponding master public key, and one of the two adversarially given equal-length messages (where the challenge message is determined by a uniform bit b). It is important to notice that the experiment uses the same uniform bit b across all instances. Eventually, the adversary outputs a guess on b and succeeds if its guess is correct and it is valid in the sense of (μ, q)-IBE-IND-CPA.

[2]IND-CPA [GM84] is a standard security notion for PKE schemes considering one instance and one challenge ciphertext.

[3]IND-CCA [NY90, RS92] is considered to be a strong standard security notion for PKE schemes with one instance and one challenge ciphertext.

(In Figure 4.1, we give a schematic description of the (μ, q)-IBE-IND-CPA-security experiment. See below for a formal definition.) Secondly, we give a variant of nested dual system groups (NDSG) [CW13], dubbed extended NDSG (ENDSG). NDSGs themselves are a variant of dual system groups (DSG) [CW14] which are based on the dual system framework due to Waters [Wat09]. NDSGs by Chen and Wee gave rise to prove the first IBE (almost) tightly IBE-IND-CPA-secure (i.e., in the one-instance, one-ciphertext setting) under a simple assumption. We extend their their work to end up with an IBE that is (almost) tightly (μ, q)-IBE-IND-CPA-secure (i.e., in the multi-instance, multi-ciphertext setting)[4]. Concretely, we give an efficient cryptographic reduction from the (μ, q)-IBE-IND-CPA-security of our IBE system to the hardness of a simple assumption and the security of the underlying ENDSG system. Note that in [HKS15], we additionally give ENDSG instantiations under simple (dual system) assumptions. However, this chapter focuses solely on the ENDSG abstraction and its tightness guarantees, not on the ENDSG instantiations.

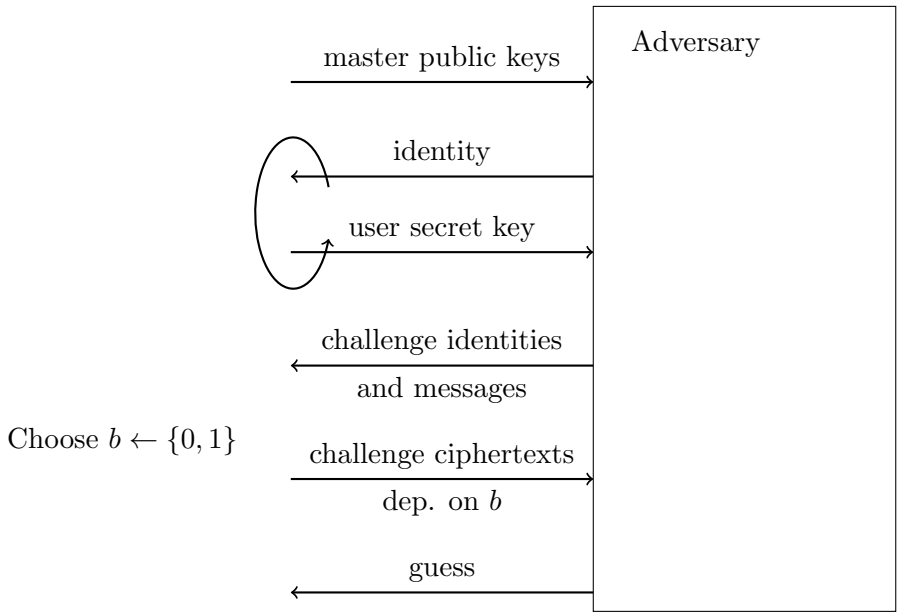

Figure 4.1: Schematic description of the (μ, q)-IBE-IND-CPA-security experiment.

[4]Technically, we need an additional simple assumption to prove "full" (μ, q)-IBE-IND-CPA-security. See below for details.

The Approach of Chen and Wee and Why an Extension to the (μ, q)-IBE-IND-CPA-setting is not Obvious. We explain the (high-level) proof strategy of [CW13] in the IBE-IND-CPA-security scenario and then turn to the reduction strategy we use to proof our IBE (almost) tightly (μ, q)-IBE-IND-CPA-secure. (Our high-level proof strategy is very similar to Chen and Wee's strategy while the lower-level strategy significantly deviates.) Reduction strategies are often organized in stages or "games," where one starts with an appropriate game and then "properly" introducing non-significantly notable changes between games until the adversary is no longer able to succeed in a game with probability significantly larger than the adversary's probability of guessing the challenge bit. The strategy of Chen and Wee is as follows:

- Chen and Wee start with the IBE-IND-CPA-security experiment.
- First change: the challenge ciphertext uses the master secret key msk explicitly and is "pseudo-random" such that the challenge message is "blinded" by a (pairing) term that contains a value $R(\varepsilon)$, for a truly random function R and for ε that depends on msk. (Note that the adversary gains no information about $R(\varepsilon)$ in the public parameters or in the master public key.) Further, the pairing term that contains $R(\varepsilon)$ is uniformly distributed in an appropriate group due to the properties of the underlying NDSG.
- n "hybrid" changes: let $n = n(k) \in \mathbb{N}$ be the identity length. For $i \in [n]$, the challenge ciphertext is pseudo-random of type i such that the challenge message is blinded with term that contains $R(id^*|_i)$, where $id^*|_i$ is the i-th bit prefix of the challenge identity id^*. User secret keys in the i-th hybrid for an id contain $R(id|_i)$. (Note that for $i = n$, the challenge ciphertext solely depends on id^*.)
- Last change: the challenge message is replaced by a uniform element from the message space.

The crucial point here is within the hybrid changes. Chen and Wee guess the i-th bit of the challenge identity id_i^* and can simulate the challenge ciphertext if the guess was correct. Further, for key extraction queries for an identity id such that $id_i = id_i^*$ (i.e., the i-th bit of id equals the i-th bit of id^*),

the user secret keys carry the value $R(id|_{i-1})$ (as in the $(i-1)$-th hybrid) while for $id_i \neq id_i^*$, the user secret keys carry the value $R(id|_{i-1}) \cdot R'(id|_{i-1})$. Depending on the input of the reduction (where the computational challenge is embedded in R'), it holds that $R'(id_{i-1}) = 1$ (as in the $(i-1)$-th hybrid), for all id, or R' is truly random function (as in the i-th hybrid). After n hybrid changes, each user secret key for id depends on $R(id)$ while the challenge ciphertext depends on $R(id^*)$. Concerning (μ, q)-IBE-IND-CPA-security, since Chen and Wee have to guess the challenge identity bit between the hybrids, it is not clear how to extend their result to a setting with multiple challenge identities (and, thus, challenge ciphertexts) or instances.

Our Approach. We stress that at a higher level our proof approach is very similar to the proof approach of [CW13], but deviates on the lower level. We proceed as follows:

- We start with the (μ, q)-IBE-IND-CPA-security experiment.
- First change: all challenge ciphertexts are pseudo-random, i.e., they use explicitly the appropriate master secret keys and contain a pairing term with an $R_j(\varepsilon_j)$-element that blinds the instance-j challenge messages, for truly random functions $(R_j)_j$. This is analogously to Chen and Wee's first change, but extended to the multi-instance, multi-ciphertext case with μ independently generated master secret keys and independently sampled truly random functions. However, due to the properties of the underlying ENDSG, the random function output can be mapped into two different subgroups depending on the input to the encryption function. (This is different to Chen and Wee.) Here, the output of the random functions is mapped into the first subgroup.
- For all $i \in [n]$, we continue with $3n$ hybrid changes:

  (i.1) Depending on the i-th challenge identity bit id_i^*, the output of the random function $R_j(id^*|_{i-1})$ is mapped into two different subgroups that "influence" the challenge ciphertext differently. If $id_i^* = 0$, then $R(id^*|_{i-1})$ is mapped into the first subgroup; if $id_i^* = 1$, then $R(id^*|_{i-1})$ is mapped into the second one. Further, the user secret keys contain $R_j(id|_{i-1})$, for some identity id. (Note that $\varepsilon_j = R_j(id|_0)$ holds.)

(i.2) The challenge ciphertexts now depend on $R_j(id^*|_i)$ while the user secret keys depend on $R_j(id|_i)$, for all queried id.

(i.3) We map the $R_j(id^*|_i)$-elements back to the first subgroup.

– Last change: each challenge message is replaced by a uniform element from the message space.

4.1 Extended Nested Dual System Groups

We define extended nested dual system groups (ENDSG) as a variant of nested dual system groups (NDSG) by [CW13]. This section is reproduced and partly adopted verbatim from [HKS15].

(Nested) Dual System Groups. Chen and Wee's nested dual system groups (NDSG) [CW13] can be seen as a variant of their dual system groups (DSG) [CW14] which itself is based on the dual system framework introduced by Waters in [Wat09]. Recently, NDSGs gave rise to prove the first IBE (almost) tightly IBE-IND-CPA-secure under simple assumptions.

A Variant of Nested Dual System Groups. We introduce a variant of NDSGs, dubbed extended NDSGs (ENDSG). (Mainly, we re-use and extend the notions from [CW13].) Let $\mathsf{G}(k, n')$ be a group generator that, given integers k and n' (where n' is a constant and, in particular, independent of k), generates the tuple

$$(G, H, G_T, N, (g_{p_1}, \ldots, g_{p_{n'}}), (h_{p_1}, \ldots, h_{p_{n'}}), g, h, e),$$

for composite-order groups G, H, G_T, all of known group order $N = p_1 \cdots p_{n'}$, for k-bit primes $(p_i)_i$, and for a pairing $e : G \times H \to G_T$. Further, g and h are generators of G and H, and $(g_{p_i})_i$ and $(h_{p_i})_i$ are generators of the (proper) subgroups $G_{p_i} \subset G$ and $H_{p_i} \subset H$ of order $|G_{p_i}| = |H_{p_i}| = p_i$, respectively. In this setting, an ENDSG ENDSG consists of the PPT algorithms $\mathsf{SampP}, \mathsf{SampG}, \mathsf{SampH}, \widetilde{\mathsf{SampG}}, \widetilde{\mathsf{SampG}}$:

Parameter Sampling. $\mathsf{SampP}(k, n)$, given security parameter k and parameter $n \in \mathbb{N}$, samples

$$(G, H, G_T, N, (g_{p_1}, \ldots, g_{p_{n'}}), (h_{p_1}, \ldots, h_{p_{n'}}), g, h, e) \leftarrow \mathsf{G}(k, n'),$$

for a constant integer n' determined by SampP, and outputs public and secret parameters

$$pp = (G, H, G_T, N, g, h, e, m, n, pars) \text{ and } sp = (\widehat{h}, \widetilde{h}, \widehat{pars}, \widetilde{pars}),$$

respectively, where $m : H \to G_T$ is a linear map, $\widehat{h}, \widetilde{h}$ are nontrivial H-elements, and $pars, \widehat{pars}, \widetilde{pars}$ may contain arbitrary additional information used by $\mathsf{SampG}, \mathsf{SampH}$, and $\widehat{\mathsf{SampG}}$ and $\widetilde{\mathsf{SampG}}$.

G-group Sampling. $\mathsf{SampG}(pp)$, given pp, outputs

$$\mathbf{g} = (g_0, \ldots, g_n) \in G^{n+1}.$$

H-group Sampling. $\mathsf{SampH}(pp)$, given pp, outputs

$$\mathbf{h} = (h_0, \ldots, h_n) \in H^{n+1}.$$

Semi-functional G-group Sampling 1. $\widehat{\mathsf{SampG}}(pp, sp)$, given pp and sp, outputs

$$\widehat{\mathbf{g}} = (\widehat{g}_0, \ldots, \widehat{g}_n) \in G^{n+1}$$

Semi-functional G-group Sampling 2. $\widetilde{\mathsf{SampG}}(pp, sp)$, given pp and sp, outputs

$$\widetilde{\mathbf{g}} = (\widetilde{g}_0, \ldots, \widetilde{g}_n) \in G^{n+1}.$$

We define correctness and security of an ENDSG system ENDSG in the following sense:

Correctness of ENDSG. For all $k \in \mathbb{N}$, for all integers $n = n(k) > 1$, for all pp, where pp is the first output of $\mathsf{SampP}(k, n)$, we require:

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 that

$$e(g_0, h_i) = e(g_i, h_0)$$

holds, for all $i \in [n]$.

Projective. For all $s \leftarrow \mathbb{Z}_N^*$, for all g_0 as the first output of $\mathsf{SampG}(pp; s)$, for all $h \in H$, we have that

$$m(h)^s = e(g_0, h).$$

Security of ENDSG. For all $k \in \mathbb{N}$, for all integers $n = n(k) > 1$, for all $(pp, sp) \leftarrow \mathsf{SampP}(k, n)$, we require:

Orthogonality. For m specified in pp, for $\widehat{h}, \widetilde{h}$ specified in sp, we have

$$m(\widehat{h}) = m(\widetilde{h}) = 1.$$

For the group elements 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

$$e(g_0, \widehat{h}) = 1,\ e(g_0, \widetilde{h}) = 1,\ e(\widehat{g}_0, \widetilde{h}) = 1,\ \text{and}\ e(\widetilde{g}_0, \widehat{h}) = 1.$$

G- and H-subgroups. The outputs of SampG, $\widehat{\mathsf{SampG}}$, and $\widetilde{\mathsf{SampG}}$ are distributed uniformly over the generators of different nontrivial subgroups of 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 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 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 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). Left-subgroup indistinguishability 1 requires that for any PPT adversary D, the function

$$\mathsf{Adv}^{\mathrm{ls1}}_{\mathsf{ENDSG},\mathsf{G},D}(k, n) := |\Pr[D(pp, \mathbf{g}) = 1] - \Pr[D(pp, \mathbf{g}\widehat{\mathbf{g}}) = 1]|$$

is negligible in k, where $\mathbf{g} \leftarrow \mathsf{SampG}(pp)$ and $\widehat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(pp, sp)$.

Left-subgroup Indistinguishability 2 (LS2). Left-subgroup indistinguishability 2 requires that for any PPT adversary D, the function

$$\begin{aligned}\mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG},\mathsf{G},D}(k,n) := |\Pr\left[D(pp,\widehat{\widetilde{h}}\widetilde{h},\mathbf{g}'\widehat{\mathbf{g}}',\mathbf{g}\widehat{\mathbf{g}})=1\right] \\ -\Pr\left[D(pp,\widehat{\widetilde{h}}\widetilde{h},\mathbf{g}'\widehat{\mathbf{g}}',\mathbf{g}\widetilde{\mathbf{g}})=1\right]|\end{aligned}$$

is negligible in k, where $\mathbf{g},\mathbf{g}' \leftarrow \mathsf{SampG}(pp)$, $\widehat{\mathbf{g}},\widehat{\mathbf{g}}' \leftarrow \widehat{\mathsf{SampG}}(pp,sp)$, and $\widetilde{\mathbf{g}} \leftarrow \widetilde{\mathsf{SampG}}(pp,sp)$, for $\widehat{h}$ and $\widetilde{h}$ specified in sp.

Nested-hiding Indistinguishability (NH). Nested-hiding indistinguishability requires that for any PPT adversary D, for all integers $q' = q'(k)$, the function

$$\begin{aligned}&\mathsf{Adv}^{\mathrm{nh}}_{\mathsf{ENDSG},\mathsf{G},D}(k,n,q') := \\ &\max_{i\in[\lfloor\frac{n}{2}\rfloor]}\Big(|\Pr\left[D(pp,\widehat{h},\widetilde{h},\widehat{\mathbf{g}}_{-(2i-1)},\widetilde{\mathbf{g}}_{-2i},(\mathbf{h}_1,\ldots,\mathbf{h}_{q'}))=1\right] \\ &-\Pr\left[D(pp,\widehat{h},\widetilde{h},\widehat{\mathbf{g}}_{-(2i-1)},\widetilde{\mathbf{g}}_{-2i},(\mathbf{h'}_1,\ldots,\mathbf{h'}_{q'}))=1\right]|\Big),\end{aligned}$$

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 $\widehat{h},\widetilde{h}$ specified in sp, for $\widehat{\gamma}_{i'},\widetilde{\gamma}_{i'} \leftarrow \mathbb{Z}^*_{\mathsf{ord}(H)}$, and for all $i' \in [q']$.

Informal Comparison of NDSGs and ENDSGs. Loosely speaking, in contrast to the NDSGs [CW13], ENDSGs have a second semi-functional G-group sampling algorithm $\widetilde{\mathsf{SampG}}$ as well as a second nontrivial H-element in sp (i.e., $\widetilde{h}$). Further, the SampGT-algorithm is omitted in ENDSGs. Concerning the ENDSG properties, we extend the NDSG properties appropriately and introduce one additional property (i.e., LS2).

4.2 An (Almost) Tightly Secure IBE

We are now ready to construct an (almost) tightly (μ, q)-IBE-IND-CPA-secure IBE from an ENDSG system and a universal hash function. (This section is reproduced and partly adopted verbatim from [HKS15].) We define (weak) (μ, q)-IBE-IND-CPA-security as follows:

(Weak) (μ, q)-IBE-IND-CPA-security. We say an IBE scheme is (μ, q)-IBE-IND-CPA-secure if and only if any PPT adversary A has only negligible advantage in the following security experiment. Let

$$\mathsf{Enc}'(pp, mpk, id, b, M_0, M_1)$$

be a PPT auxiliary encryption oracle that, given pp and mpk, a (challenge) identity $id \in \mathcal{ID}$, $b \in \{0,1\}$, and two (challenge) messages $M_0, M_1 \in \mathcal{M}$, outputs a (challenge) ciphertext $C_{id} \leftarrow \mathsf{Enc}(mpk, id, M_b)$. First, A gets honestly generated public parameter pp and master public keys $(mpk_j)_j$, for all $j \in [\mu]$. During the experiment, A has access to $\mathsf{Ext}(msk_j, \cdot)$-oracles and $\mathsf{Enc}'(pp, mpk_j, \cdot, b, \cdot, \cdot)$-oracles, for $b \leftarrow \{0,1\}$ and for all $j \in [\mu]$. Eventually, A outputs a guess b'. We say that A is valid if

- it has not queried its instance-j Enc'-oracles and Ext-oracles with the same identity,
- it has only provided equal-length messages as input to Enc', and
- it has only queried Enc' at most q times per instance.

Finally, if $b = b'$ and A is valid, then the experiment outputs 1.

More formally, we define the advantage function for an adversary A as

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

where the experiment $\mathsf{Exp}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n)$ is given in Figure 4.2 and IBE is an IBE as above. Then we say IBE is (μ, q)-IBE-IND-CPA-secure if and only if for any PPT adversary A the function $\mathsf{Adv}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k,n)$ is negligible in k.

Further, we say an IBE scheme IBE is weakly (μ, q)-IBE-IND-CPA-secure if and only if for any PPT weak adversary A the function $\mathsf{Adv}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k, n)$ is negligible in k. In our case, a weak adversary does not query its Enc'-oracle twice per identity and instance.

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

Figure 4.2: (μ, q)-IBE-IND-CPA experiment for IBE schemes.

A Variant of Chen and Wee's IBE. We are now ready to present our variant of Chen and Wee's IBE scheme from [CW13]. As a basic building block, we use an ENDSG $\mathsf{ENDSG} = (\mathsf{SampP}, \mathsf{SampG}, \mathsf{SampH}, \widehat{\mathsf{SampG}}, \widetilde{\mathsf{SampG}})$ from Section 4.1. Besides, for groups G_T (defined below), let $\mathcal{UH}$ be a family of universal hash functions $\mathsf{H} : G_T \to \{0,1\}^k$ such that for any nontrivial subgroup $G'_T \subset G_T$, and for $\mathsf{H} \leftarrow \mathcal{UH}$, $X \leftarrow G'_T$, and $U \leftarrow \{0,1\}^k$, we have $\mathsf{SD}\left((\mathsf{H}, \mathsf{H}(X)); (\mathsf{H}, U)\right) = \mathbf{O}(2^{-k})$. Let an IBE $\mathsf{IBE} = (\mathsf{Par}, \mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ with identity space $\mathcal{ID} = \{0,1\}^n$, for integer $n = n(k)$, and message space $\mathcal{M} = \{0,1\}^k$, be defined as follows:

Parameter generation. $\mathsf{Par}(k, n)$, given k and n, samples $(pp', sp') \leftarrow \mathsf{SampP}(k, 2n)$, for

$$pp' = (G, H, G_T, N, g, h, e, m, 2n, pars) \text{ and}$$

$$sp' = (\widehat{h}, \widetilde{h}, \widehat{pars}, \widetilde{pars}).$$

Further, Par samples $\mathsf{H} \leftarrow \mathcal{UH}$ and outputs the public and secret parameters (pp, sp), where $pp = (pp', \mathsf{H})$ and $sp = sp'$.

Key generation. $\mathsf{Gen}(pp, sp)$, given parameters pp and sp, samples $msk \leftarrow H$ and outputs a master public key $mpk := (pp, m(msk))$ and a master secret key msk.

Secret-key extraction. $\mathsf{Ext}(msk, id)$, given $msk \in H$ 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

$$usk_{id} := (h_0, msk \cdot \prod_{i=1}^{n} h_{2i-id_i}).$$

Encryption. $\mathsf{Enc}(mpk, id, M)$, given $mpk = (pp, m(msk))$, an identity $id = (id_1 \dots id_n) \in \mathcal{ID}$, and a message $M \in \mathcal{M}$, computes

$$(g_0, \dots, g_{2n}) := \mathsf{SampG}(pp; s),$$

for $s \leftarrow \mathbb{Z}_N^*$, and $g_T := m(msk)^s$ $(= e(g_0, msk))$, and outputs a ciphertext

$$C_{id} := (g_0, \prod_{i=1}^{n} g_{2i-id_i}, \mathsf{H}(g_T) \oplus M).$$

Decryption. $\mathsf{Dec}(usk_{id}, C_{id'})$, given a user secret key $usk_{id} =: (K_0, K_1)$ and a ciphertext $C_{id'} =: (C_0, C_1, C_2)$, outputs

$$M := \mathsf{H}\left(\frac{e(C_0, K_1)}{e(C_1, K_0)}\right) \oplus C_2.$$

We show correctness and (almost) tight (μ, q)-IBE-IND-CPA-security of IBE as follows:

Correctness of IBE. We have

$$\begin{aligned} & \mathsf{H}\left(\frac{e(C_0, K_1)}{e(C_1, K_0)}\right) \oplus C_2 \\ = \ & \mathsf{H}\left(\frac{e(g_0, msk \cdot \prod_{i=1}^{n} h_{2i-id_i})}{e(\prod_{i=1}^{n} g_{2i-id'_i}, h_0)}\right) \oplus \mathsf{H}(g_T) \oplus M \\ \overset{(*)}{=} \ & \mathsf{H}(g_T) \oplus \mathsf{H}(g_T) \oplus M, \end{aligned}$$

for $id = id'$. Note that $(*)$ holds due to ENDSG's associativity and projective properties.

(μ, q)-IBE-IND-CPA-security of IBE. We base our high-level proof strategy on the IBE-IND-CPA proof strategy of Chen and Wee [CW13], but deviate on the lower level. First, we define 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]:

Auxiliary secret-key extraction. $\overline{\mathsf{Ext}}(pp, msk, id; \mathbf{h})$, given the parameter pp, master secret key msk, an identity $id = id_1 \dots id_n \in \mathcal{ID}$, and $\mathbf{h} = (h_0, \dots, h_{2n}) \in (H)^{2n+1}$, outputs a user secret key

$$usk_{id} := (h_0, msk \cdot \prod_{i=1}^{n} h_{2i-id_i}).$$

Auxiliary encryption function. $\overline{\mathsf{Enc}}(pp, id, M; msk, \mathbf{g})$, given parameter pp, identity $id = id_1 \dots id_n \in \mathcal{ID}$, message $M \in \mathcal{M}$, master secret key msk, and $\mathbf{g} = (g_0, \dots, g_{2n}) \in (G)^{2n+1}$, 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 $id|_i := id_1 \dots id_i$ be the i-bit prefix of an identity id, and let $\mathcal{ID}|_i := \{0,1\}^i$. For an instance j and $i \in [n] \cup \{0\}$, consider functions

$$\widehat{\mathsf{R}}_{j,i} : \mathcal{ID}|_i \to H,\ id|_i \mapsto (\widehat{h})^{\widehat{\gamma}_{j,i}(id|_i)} \quad \text{and}$$

$$\widetilde{\mathsf{R}}_{j,i} : \mathcal{ID}|_i \to H,\ id|_i \mapsto (\widetilde{h})^{\widetilde{\gamma}_{j,i}(id|_i)},$$

where

$$\widehat{\gamma}_{j,i} : \mathcal{ID}|_i \to \mathbb{Z}^*_{\mathsf{ord}(H)}, id|_i \mapsto \widehat{\gamma}_{j,id|_i} \quad \text{and}$$

$$\widetilde{\gamma}_{j,i} : \mathcal{ID}|_i \to \mathbb{Z}^*_{\mathsf{ord}(H)}, id|_i \mapsto \widetilde{\gamma}_{j,id|_i}$$

are independently and truly random functions.

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

$$\mathbf{g} = (g_0, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(pp) \quad \text{and}$$
$$\widehat{\mathbf{g}} = (\widehat{g}_0, \ldots, \widehat{g}_{2n}) \leftarrow \widehat{\mathsf{SampG}}(pp, sp).$$

(Hence, pseudo-normal ciphertexts have G-components sampled from $\widehat{\mathsf{SampG}}$.)

Semi-functional type-$(\wedge, i)$ and type-$(\sim, i)$ ciphertexts. We define the random functions $\widehat{\mathsf{R}}_{j,i}$ and $\widetilde{\mathsf{R}}_{j,i}$ as above. The 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

$$\mathbf{g} = (g_0, \ldots, g_{2n}) \leftarrow \mathsf{SampG}(pp),$$
$$\widehat{\mathbf{g}} = (\widehat{g}_0, \ldots, \widehat{g}_{2n}) \leftarrow \widehat{\mathsf{SampG}}(pp), \quad \text{and}$$
$$\widetilde{\mathbf{g}} = (\widetilde{g}_0, \ldots, \widetilde{g}_{2n}) \leftarrow \widetilde{\mathsf{SampG}}(pp),$$

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 4.2.1. *If* ENDSG *is an ENDSG system as defined in Section 4.1 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 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*

$$\begin{aligned} \mathsf{Adv}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k, n) \leq\ & \mathsf{Adv}^{\mathrm{ls1}}_{\mathsf{ENDSG},\mathsf{G},D_1}(k, 2n) + 2n \cdot \mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG},\mathsf{G},D_2}(k, 2n) \\ & + n \cdot \mathsf{Adv}^{\mathrm{nh}}_{\mathsf{ENDSG},\mathsf{G},D_3}(k, 2n, \mu q') + \mu q \cdot \mathbf{O}(2^{-k}), \quad (4.1) \end{aligned}$$

for group generator G *defined as above.*

Proof. We show the (μ, q)-IBE-IND-CPA security of IBE for any weak PPT adversary A in a sequence of games where we successively change the games until we arrive at a game where A has only negligible advantage (i.e., success probability only negligibly larger than $1/2$) in the sense of (μ, q)-IBE-IND-CPA. Let $S_{A,j}$ be the event that A succeeds in Game j. In Table 4.1, we give an overview how the challenge ciphertexts and user secret keys are generated.

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 and only 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 and only 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 the user secret keys are semi-functional of type i.

Game 3. Game 3 is defined as Game $2.n.0$ except that the challenge ciphertexts are semi-functional of type $(\wedge, n)$ and the user secret keys are semi-functional of type n.

Game 4. Game 4 is defined as Game 3 except that the challenge ciphertext messages are uniform k-length bitstrings.

Lemma 4.2.2 (Game 0 − Game 1). *If the G- and H-subgroups property and LS1 of* ENDSG *hold, Game 0 and Game 1 are computationally indistinguishable. 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 is a distinguisher D on LS1 with running time $t' \approx t + \mathbf{O}(\mu n k^c (q + q'))$, for some constant $c \in \mathbb{N}$, such that*

$$|\Pr[S_{A,0}] - \Pr[S_{A,1}]| \leq \mathsf{Adv}^{\mathrm{ls1}}_{\mathsf{ENDSG},\mathsf{G},D}(k, 2n), \tag{4.2}$$

for group generator G *defined as above.*

Proof. In Game 0, all challenge ciphertexts are normal in the sense of IBE while in Game 1, all challenge ciphertexts are pseudo-normal. In the following, we give a description and an analysis of an efficient LS1 distinguisher that uses any efficient IBE-attacker in the (μ, q)-IBE-IND-CPA sense.

Description. The challenge input is provided as $(pp, \mathbf{T})$, where $\mathbf{T}$ is either

$$\mathbf{g} \quad \text{or} \quad \mathbf{g}\widehat{\mathbf{g}},$$

for $\mathbf{g} \leftarrow \mathsf{SampG}(pp)$, $\widehat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(pp, sp)$, and

$$pp = (G, H, G_T, N, g, h, e, m, 2n, pars).$$

	Challenge ciphertexts for $id^*_{j,i'}$
Game 0	$\mathsf{Enc}(mpk_j, id^*_{j,i'}, M^*_{j,i',b})$
Game 1	$\underline{\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j, \mathbf{g}\widehat{\mathbf{g}})}$
Game 2.i.0	$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,i-1}(id^*_{j,i'}\vert_{i-1})}, \mathbf{g}\widehat{\mathbf{g}})$
Game 2.i.1	if $id^*_{j,i',i} = 0 : \overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \widehat{\mathsf{R}}_{j,i-1}(id^*_{j,i'}\vert_{i-1}), \mathbf{g}\widehat{\mathbf{g}})$
	if $id^*_{j,i',i} = 1 : \overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widetilde{\mathsf{R}}_{j,i-1}(id^*_{j,i'}\vert_{i-1})}, \underline{\mathbf{g}\widetilde{\mathbf{g}}})$
Game 2.i.2	if $id^*_{j,i',i} = 0 : \overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,i}(id^*_{j,i'}\vert_{i})}, \mathbf{g}\widehat{\mathbf{g}})$
	if $id^*_{j,i',i} = 1 : \overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widetilde{\mathsf{R}}_{j,i}(id^*_{j,i'}\vert_{i})}, \mathbf{g}\widetilde{\mathbf{g}})$
Game 3	$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,n}(id^*_{j,i'})}, \mathbf{g}\widehat{\mathbf{g}})$
Game 4	$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, \underline{R_{j,i'}}; msk_j \cdot \widehat{\mathsf{R}}_{j,n}(id^*_{j,i'}), \mathbf{g}\widehat{\mathbf{g}})$
	User secret keys for id
Game 0	$\mathsf{Ext}(msk_j, id)$
Game 1	$\underline{\overline{\mathsf{Ext}}(pp, msk_j, id; \mathbf{h})}$
Game 2.i.0	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,i-1}(id\vert_{i-1}) \cdot \widetilde{\mathsf{R}}_{j,i-1}(id\vert_{i-1})}, id; \mathbf{h})$
Game 2.i.1	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}_{j,i-1}(id\vert_{i-1}) \cdot \widetilde{\mathsf{R}}_{j,i-1}(id\vert_{i-1}), id; \mathbf{h})$
Game 2.i.2	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,i}(id\vert_{i}) \cdot \widetilde{\mathsf{R}}_{j,i}(id\vert_{i})}, id; \mathbf{h})$
Game 3	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \underline{\widehat{\mathsf{R}}_{j,n}(id) \cdot \widetilde{\mathsf{R}}_{j,n}(id)}, id; \mathbf{h})$
Game 4	$\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}_{j,n}(id) \cdot \widetilde{\mathsf{R}}_{j,n}(id), id; \mathbf{h})$

Table 4.1: 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.

First, D samples $(msk_j)_j \leftarrow (H)^\mu$, sets $mpk_j := (pp, \mathsf{H}, m(msk_j))$, for all j, for $\mathsf{H} \leftarrow \mathcal{UH}$, and sends $(mpk_j)_j$ to A. During the experiment, D answers instance-j secret key extraction queries to oracle $\mathsf{Ext}(msk_j, \cdot)$, for $id \in \mathcal{ID}$, as

$$\overline{\mathsf{Ext}}(pp, msk_j, id; \mathsf{SampH}(pp)),$$

for all j. (We assume that A queries at most q' user secret keys per instance.) Then, D fixes a bit $b \leftarrow \{0,1\}$. A may adaptively query its Enc'-oracle; for A-chosen instance-j challenge identity $id^*_{j,i} \in \mathcal{ID}$ and equal-length messages $(M^*_{j,i,0}, M^*_{j,i,1})$. D returns

$$\overline{\mathsf{Enc}}(pp, id^*_{j,i}, M^*_{j,i,b}; msk_j, \mathbf{T}^{s_{j,i}})$$

to A, for $s_{j,i} \leftarrow \mathbb{Z}^*_N$, for all $(j,i) \in [\mu] \times [q]$. (We assume that A queries at most q challenge ciphertexts per instance.) Eventually, A outputs a guess b'. D outputs 1 if $b' = b$ and A is valid in the sense of (μ, q)-IBE-IND-CPA, else outputs 0.

Analysis. The provided master public keys and the A-requested user secret keys yield the correct distribution and are consistent in the sense of Game 0 and Game 1. Due to ENDSG's G- and H-subgroups property, we have that $\mathbf{T}$ is uniformly distributed over the generators of a nontrivial subgroup of G^{2n+1}. Hence, $\mathbf{T}^s$, for $s \leftarrow \mathbb{Z}^*_N$, is distributed uniformly over the generators of a nontrivial subgroup of G^{2n+1} and, thus, all challenge ciphertexts yield the correct distribution in the sense of Game 0 and Game 1. If $\mathbf{T} = \mathbf{g}$, then the challenge ciphertexts are distributed identically as in Game 0. Otherwise, i.e., if $\mathbf{T} = \mathbf{g}\widehat{\mathbf{g}}$, then the challenge ciphertexts are distributed identically as in Game 1. Hence, (4.2) follows. □

Lemma 4.2.3 (Game 1 – Game 2.1.0). *If the orthogonality property of* ENDSG *holds, the output distributions of Game 1 and Game 2.1.0 are the same. Concretely, for any PPT adversary A in the (μ, q)-IBE-IND-CPA security experiment with* IBE*, it holds that*

$$\Pr[S_{A,1}] = \Pr[S_{A,2.1.0}]. \tag{4.3}$$

Proof. In this bridging step, we argue that each instance-j master secret key msk_j, with $msk_j \leftarrow H$, generated as in Game 1 and the (implicit) instance-j

master secret keys msk'_j, with

$$msk'_j := msk''_j \cdot \widehat{\mathsf{R}}_{j,0}(\varepsilon) \cdot \widetilde{\mathsf{R}}_{j,0}(\varepsilon),$$

for $msk''_j \leftarrow H$ and $\widehat{\mathsf{R}}_{j,0}, \widetilde{\mathsf{R}}_{j,0}$ defined as above, generated as in Game 2.1.0, are identically distributed, for all j. Note that the master public keys for A contain $(m(msk_j))_j$; but since

$$((m(msk'_j))_j = (m(msk''_j))_j,$$

which is due to the orthogonality property of ENDSG, no $\widehat{\mathsf{R}}_{j,0}$-information and no $\widetilde{\mathsf{R}}_{j,0}$-information is given out in the master public keys. Further, since $(msk_j)_j$ and $(msk''_j)_j$ are identically distributed, it follows that (4.3) holds. □

Lemma 4.2.4 (Game 2.i.0 – Game 2.i.1). *If the G- and H-subgroups property and LS2 of* ENDSG *hold, Game 2.i.0 and Game 2.i.1 are computationally indistinguishable. 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 is a distinguisher D on LS2 with running time $t' \approx t + \mathbf{O}(\mu n k^c (q + q'))$, for some constant $c \in \mathbb{N}$, such that*

$$|\Pr[S_{2.i.0}] - \Pr[S_{2.i.1}]| \leq \mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG},\mathsf{G},D}(k, 2n), \tag{4.4}$$

for group generator G *defined as above and for all $i \in [n]$.*

Proof. In Game 2.i.0, we have semi-functional type-$(\wedge, i-1)$ challenge ciphertexts while in Game 2.i.1, challenge ciphertexts are semi-functional of type $(\sim, i-1)$ if and only if the i-th challenge identity bit is 1.

Description. The challenge input is provided as

$$(pp, \widehat{h}\widetilde{h}, \mathbf{g}'\widehat{\mathbf{g}}', \mathbf{T}),$$

where $\mathbf{T}$ is either

$$\mathbf{g}\widehat{\mathbf{g}} \quad \text{or} \quad \mathbf{g}\widetilde{\mathbf{g}},$$

for pp as before, for $\widehat{h}, \widetilde{h}$ specified in sp, for

$$\mathbf{g}, \mathbf{g}' \leftarrow \mathsf{SampG}(pp),$$

$$\widehat{\mathbf{g}}, \widehat{\mathbf{g}}' \leftarrow \widehat{\mathsf{SampG}}(pp, sp), \text{ and}$$

$$\widetilde{\mathbf{g}} \leftarrow \widetilde{\mathsf{SampG}}(pp, sp).$$

First, D samples $(msk_j)_j \leftarrow (H)^\mu$, sets $mpk_j := (pp, \mathsf{H}, m(msk_j))$, for all j, for $\mathsf{H} \leftarrow \mathcal{UH}$, for m specified in pp, and sends $(mpk_j)_j$ to A. Further, D defines a truly random function

$$\mathsf{R} : [\mu] \times \{0,1\}^{i-1} \rightarrow \langle \widehat{h}\widetilde{h} \rangle.$$

During the experiment, D answers instance-j secret key extraction queries to oracle $\mathsf{Ext}(msk_j, \cdot)$ as

$$\overline{\mathsf{Ext}}(pp, msk_j \cdot \mathsf{R}(j, id|_{i-1}), id; \mathsf{SampH}(pp)),$$

for $id \in \mathcal{ID}$ and all j. (Again, we assume that A queries at most q' user secret keys per instance and we set $id|_0 = \{0,1\}^0 =: \varepsilon$.) A may adaptively query its Enc'-oracle; for instance-j challenge identity $id^*_{j,i'} = id^*_{j,i',1} \ldots, id^*_{j,i',n} \in \mathcal{ID}$ and equal-length messages $(M^*_{j,i',0}, M^*_{j,i',1})$, D returns

$$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \mathsf{R}(j, id^*_{j,i'}|_{i-1}), (\mathbf{g}'\widehat{\mathbf{g}}')^{s_{j,i'}}) \quad \text{if } id^*_{j,i',i} = 0,$$

$$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \mathsf{R}(j, id^*_{j,i'}|_{i-1}), \mathbf{T}^{s_{j,i'}}) \quad \text{if } id^*_{j,i',i} = 1,$$

to A, for $b \leftarrow \{0,1\}$, for $s_{j,i'} \leftarrow \mathbb{Z}^*_N$, for all $(j, i') \in [\mu] \times [q]$. Eventually, A outputs a guess b'. D outputs 1 if $b' = b$ and A is valid in the sense of (μ, q)-IBE-IND-CPA, else outputs 0.

Analysis. The master public keys yield the correct distribution as well as the requested user secret keys (which is due to ENDSG's G- and H-subgroups property, i.e., the output of SampH is uniformly distributed over the generators of a nontrivial subgroup of H^{2n+1}). For the challenge ciphertexts, note that $\mathbf{g}'\widehat{\mathbf{g}}'$ and $\mathbf{T}$ are uniformly distributed over the generators of their respective nontrivial subgroup of G^{2n+1} and, hence, $(\mathbf{g}'\widehat{\mathbf{g}}')^s$ and $\mathbf{T}^s$, for $s \leftarrow \mathbb{Z}^*_N$, are distributed uniformly over the generators of their respective nontrivial G^{2n+1}-subgroup as well. If $\mathbf{T} = \mathbf{g}\widehat{\mathbf{g}}$, then the challenge ciphertexts are distributed

identically as in Game 2.i.0. Otherwise, if $\mathbf{T} = \mathbf{g}\widetilde{\mathbf{g}}$, then the challenge ciphertexts are distributed identically as in Game 2.i.1 (where, in both cases, ENDSG's orthogonality and non-degeneracy properties hold; thus, $\widehat{h}$ and $\widetilde{h}$ must contain coprime nontrivial elements and the challenge ciphertexts yield the correct distribution). Hence, (4.4) follows. □

Lemma 4.2.5 (Game 2.i.1 – Game 2.i.2). *If the G- and H-subgroups property and NH of* ENDSG *hold, Game 2.i.1 and Game 2.i.2 are computationally indistinguishable. 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 is a distinguisher D on NH with running time $t' \approx t + \mathbf{O}(\mu n k^c (q + q'))$, for some constant $c \in \mathbb{N}$, such that*

$$|\Pr[S_{2.i.1}] - \Pr[S_{2.i.2}]| \leq \mathsf{Adv}^{\mathrm{nh}}_{\mathsf{ENDSG},\mathsf{G},D}(k, 2n, \mu q'), \tag{4.5}$$

for group generator G *defined as above and for all $i \in [n]$.*

Proof. In Game 2.i.1, the challenge ciphertexts are semi-functional of type $(\wedge, i-1)$ if the i-th bit of the challenge identity is 0 and semi-functional of type $(\sim, i-1)$ if the i-th bit of the challenge identity is 1, while in Game 2.i.2, all challenge ciphertexts are of type $(\cdot, i)$ (where $\cdot$ can be $\wedge$ or $\sim$, depending on the i-th bit of the respective challenge identity).

Description. The challenge input is

$$(pp, \widehat{h}, \widetilde{h}, \widehat{\mathbf{g}}_{-(2i-1)}, \widetilde{\mathbf{g}}_{-2i}, (\mathbf{T}_{1,1}, \ldots, \mathbf{T}_{\mu,q'})),$$

where $\mathbf{T}_{j,i'}$ equals either

$$(h_{j,i',0}, \ldots, h_{j,i',2n}) \quad \text{or} \quad (h_{j,i',0}, \ldots, h_{j,i',2i-1} \cdot (\widehat{h})^{\widehat{\gamma}_{j,i'}}, h_{j,i',2i} \cdot (\widetilde{h})^{\widetilde{\gamma}_{j,i'}}, \ldots, h_{j,i',2n}),$$

for pp as before, $\widehat{h}, \widetilde{h}$ specified as in sp, for

$$\widehat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(pp, sp),$$

$$\widetilde{\mathbf{g}} \leftarrow \widetilde{\mathsf{SampG}}(pp, sp),$$

$$(h_{j,i',0}, \ldots, h_{j,i',2n}) \leftarrow \mathsf{SampH}(pp), \text{ and}$$

$$\widehat{\gamma}_{j,i'}, \widetilde{\gamma}_{j,i'} \leftarrow \mathbb{Z}^*_{\mathsf{ord}(H)},$$

for all $(j, i') \in [\mu] \times [q']$. D samples $(msk_j)_j \leftarrow (H)^\mu$, sets

$$mpk_j := (pp, \mathsf{H}, m(msk_j)),$$

for all j, for $\mathsf{H} \leftarrow \mathcal{UH}$, for m specified in pp, and sends $(mpk_j)_j$ to A. Further, D defines random functions $\widehat{\mathsf{R}}_{j,i-1}, \widetilde{\mathsf{R}}_{j,i-1}$ as above. In addition, for identity $id = id_1 \ldots id_n \in \mathcal{ID}$, we will define

$$\widehat{\mathsf{R}}_{j,i}(id|_i) := \widehat{\mathsf{R}}_{j,i-1}(id|_{i-1}) \text{ and (implicitly) } \widetilde{\mathsf{R}}_{j,i}(id|_i) := \widetilde{\mathsf{R}}_{j,i-1}(id|_{i-1}) \cdot (\widetilde{h})^{\widetilde{\gamma}_{j,i'}}$$

if $id_i = 0$ and

$$\widetilde{\mathsf{R}}_{j,i}(id|_i) := \widetilde{\mathsf{R}}_{j,i-1}(id|_{i-1}) \text{ and (implicitly) } \widehat{\mathsf{R}}_{j,i}(id|_i) := \widehat{\mathsf{R}}_{j,i-1}(id|_{i-1}) \cdot (\widehat{h})^{\widehat{\gamma}_{j,i'}}$$

if $id_i = 1$, for suitable $(j, i') \in [\mu] \times [q']$ as shown below. Further, during the experiment, D returns the i'-th secret key extraction query in instance j for an identity id, with prefix $id|_i$ not a prefix of an already queried identity in instance j, as

$$\begin{aligned}
&\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}_{j,i}(id|_i) \cdot \widetilde{\mathsf{R}}_{j,i-1}(id|_{i-1}), id; \mathbf{T}_{j,i'}) \quad \text{if } id_i = 0,\\
&\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}_{j,i-1}(id|_{i-1}) \cdot \widetilde{\mathsf{R}}_{j,i}(id|_i), id; \mathbf{T}_{j,i'}) \quad \text{if } id_i = 1,
\end{aligned}$$

for all (j, i'). (Note that $id|_i$ could be a valid prefix in any other instance different to j. Further, we assume that A queries at most q' user secret keys per instance.) For an identity prefixes $id|_i$ that is a prefix of an already queried identity in instance j, let $(j, i'') \in [\mu] \times [q']$ be the index of that query. In that case, D returns

$$\begin{aligned}
&\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}_{j,i}(id|_i) \cdot \widetilde{\mathsf{R}}_{j,i-1}(id|_{i-1}), id; \mathbf{T}_{j,i''} \cdot \mathsf{SampH}(pp)) \quad \text{if } id_i = 0,\\
&\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}_{j,i-1}(id|_{i-1}) \cdot \widetilde{\mathsf{R}}_{j,i}(id|_i), id; \mathbf{T}_{j,i''} \cdot \mathsf{SampH}(pp)) \quad \text{if } id_i = 1,
\end{aligned}$$

for all j. (Note that we use SampH to rerandomize the H^{2n+1}-subgroup element of $\mathbf{T}_{j,i''}$.) Further, A may adaptively query its Enc'-oracle; for A-chosen instance-j challenge identity $id^*_{j,i'''} = id^*_{j,i''',1} \ldots, id^*_{j,i''',n} \in \mathcal{ID}$ and equal-length messages $(M^*_{j,i''',0}, M^*_{j,i''',1})$ and returns

$$\overline{\mathsf{Enc}}(pp, id^*_{j,i'''}, M^*_{j,i''',b}; msk_j \cdot \widehat{\mathsf{R}}_{j,i}(id^*_{j,i'''}|_i), (\mathbf{g}_{-(2i-1)}\widehat{\mathbf{g}}_{-(2i-1)})^{s_{j,i'''}}) \text{ if } id^*_{j,i''',i} = 0,$$

$$\overline{\mathsf{Enc}}(pp, id^*_{j,i'''}, M^*_{j,i''',b}; msk_j \cdot \widetilde{\mathsf{R}}_{j,i}(id^*_{j,i'''}|_i), (\mathbf{g}_{-2i}\widetilde{\mathbf{g}}_{-2i})^{s_{j,i'''}}) \text{ if } id^*_{j,i''',i} = 1,$$

to A, for $s_{j,i'''} \leftarrow \mathbb{Z}_N^*$, for $\mathbf{g} \leftarrow \mathsf{SampG}(pp)$, for fixed $b \leftarrow \{0,1\}$, for all (j, i'''). (Note that a modified $\overline{\mathsf{Enc}}$-input is provided with only $4n$ instead of $4n+2$ elements. Nevertheless, the omitted elements are not needed to generate a valid ciphertext (since it is consistent with the challenge identities $(id^*_{j,i'''})_{j,i'''}$). Hence, we assume that $\overline{\mathsf{Enc}}$ works as desired.) Eventually, A outputs a guess b'. D outputs 1 if $b' = b$ and A is valid in the sense of (μ, q)-IBE-IND-CPA, else outputs 0.

Analysis. Note that the provided master public keys yield the correct distribution. For the A-requested user secret keys, we have that since $\widehat{h}$ and $\widetilde{h}$ have nontrivial H-elements of coprime order (again, this is due to ENDSG's orthogonality and non-degeneracy properties), the random functions $\widehat{\mathsf{R}}_{j,i-1}, \widehat{\mathsf{R}}_{j,i}$ and $\widetilde{\mathsf{R}}_{j,i-1}, \widetilde{\mathsf{R}}_{j,i}$ yield the correct distributions in the sense of Game 2.i.1 and Game 2.i.2, respectively. Due to the G- and H-subgroups property of ENDSG, $\mathbf{g}_{-(2i-1)}$ and $\widehat{\mathbf{g}}_{-(2i-1)}$ as well as $\mathbf{g}_{-2i}$ and $\widetilde{\mathbf{g}}_{-2i}$ are uniformly distributed over the generators of their respective nontrivial subgroups of G^{2n} and, thus, $(\mathbf{g}_{-(2i-1)}\widehat{\mathbf{g}}_{-(2i-1)})^s$ and $(\mathbf{g}_{-2i}\widetilde{\mathbf{g}}_{-2i})^s$, for $s \leftarrow \mathbb{Z}_N^*$, are distributed uniformly over the generators of their respective nontrivial subgroup of G^{2n}. Further, if $id^*_{j,i''',i} = 0$, then it holds that $\widehat{\mathsf{R}}_{j,i}(id^*_{j,i'''}|_i) = \widehat{\mathsf{R}}_{j,i-1}(id^*_{j,i'''}|_{i-1})$ and all required components $\widehat{\mathbf{g}}_{-(2i-1)}$ to create the challenge ciphertexts are given. Analogously, if $id^*_{j,i''',i} = 1$, then we have $\widetilde{\mathsf{R}}_{j,i}(id^*_{j,i'''}|_i) = \widetilde{\mathsf{R}}_{j,i-1}(id^*_{j,i'''}|_{i-1})$ and all necessary components $\widetilde{\mathbf{g}}_{-2i}$ are provided as needed. Hence, the challenge ciphertexts and user secret keys yield the correct distribution. If

$$(\mathbf{T}_{j,i'})_{j,i'} = (h_{j,i',0}, \ldots, h_{j,i',2n})_{j,i'},$$

then the user secret keys are distributed identically as in Game 2.i.1. If

$$(\mathbf{T}_{j,i'})_{j,i'} = (h_{j,i',0}, \ldots, h_{j,i',2i-1} \cdot (\widehat{h})^{\widehat{\gamma}_{j,i'}}, h_{j,i',2i} \cdot (\widetilde{h})^{\widetilde{\gamma}_{j,i'}}, \ldots, h_{j,i',2n})_{j,i'},$$

then the user secret keys are distributed identically as in Game 2.i.2. Thus, (4.5) follows. □

Lemma 4.2.6 (Game 2.i-1.2 – Game 2.i.0). *If the G- and H-subgroups property and LS2 of* ENDSG *hold, Game 2.i-1.1 and Game 2.i.0 are computationally indistinguishable. 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 is a distinguisher* D *with running time* $t' \approx t + \mathbf{O}(\mu n k^c (q + q'))$*, for some constant* $c \in \mathbb{N}$*, such that*

$$|\Pr[S_{2.i-1.2}] - \Pr[S_{2.i.0}]| \leq \mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG},\mathsf{G},D}(k, 2n), \tag{4.6}$$

for group generator G *defined as above and for all* $i \in [n] \setminus \{1\}$*.*

Proof. In Game $2.i-1.2$, challenge ciphertexts are of type $(\cdot, i-1)$ and depend on the $(i-1)$-th challenge identity bit while in Game $2.i.0$, challenge ciphertexts are of type $(\wedge, i-1)$. This proof is very similar to the proof of Lemma 4.2.4 except that the challenge ciphertexts depend on the $(i-1)$-th instead of the i-th challenge identity bit. □

Lemma 4.2.7 (Game 2.n.2 – Game 3). *If the* G*- and* H*-subgroups property and LS2 of* ENDSG *hold, Game 2.n.2 and Game 3 are computationally indistinguishable. 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 is a distinguisher* D *with running time* $t' \approx t + \mathbf{O}(\mu n k^c (q + q'))$*, for some constant* $c \in \mathbb{N}$*, such that*

$$|\Pr[S_{A,2.n.2}] - \Pr[S_{A,3}]| \leq \mathsf{Adv}^{\mathrm{ls2}}_{\mathsf{ENDSG},\mathsf{G},D}(k, 2n), \tag{4.7}$$

for group generator G *defined as above.*

Proof. It is easy to see that Game 3 and a potential Game $2.n+1.0$ would be identical. Hence, we can reassemble the proof of Lemma 4.2.6 with $i := n+1$ and (4.7) directly follows. □

Lemma 4.2.8 (Game 3 – Game 4, weak (μ, q)-IBE-IND-CPA-security). *Game 3 and Game 4 are statistically indistinguishable. Concretely, for any weak PPT adversary* A *on the* (μ, q)*-IBE-IND-CPA security of* IBE*, it holds that*

$$|\Pr[S_{A,3}] - \Pr[S_{A,4}]| \leq \mu q \cdot \mathbf{O}(2^{-k}). \tag{4.8}$$

Proof. In Game 4, we replace each challenge message $M_{j,i',b}$, for challenge bit $b \in \{0,1\}$, with a (fresh) uniformly random k-length bitstring $R_{j,i'} \leftarrow \{0,1\}^k$.

We argue with ENDSG's non-degeneracy property and the universality of H for this change. Concretely, for instance-j Game-3 challenge ciphertexts

$$\begin{aligned}&\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \widehat{\mathsf{R}}_{j,n}(id^*_{j,i'}), (\mathbf{g}\widehat{\mathbf{g}})^{s_{j,i'}})\\ &= ((g_0\widehat{g}_0)^{s_{j,i'}}, (\prod_{i=1}^{n} g_{2i-id^*_{j,i',i}}\widehat{g}_{2i-id^*_{j,i',i}})^{s_{j,i'}}, \mathsf{H}(e((g_0\widehat{g}_0)^{s_{j,i'}}, msk_j \cdot \widehat{\mathsf{R}}_{j,n}(id^*_{j,i'})))\\ &\quad \oplus M^*_{j,i',b}),\end{aligned}$$

with

$$\mathbf{g} \leftarrow \mathsf{SampG}(pp),\ \widehat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(pp, sp),$$

for $s_{j,i'} \leftarrow \mathbb{Z}^*_N$, for all $i' \in [q]$, note that

$$e((\widehat{g}_0)^{s_{j,i'}}, \widehat{\mathsf{R}}_{j,n}(id^*_{j,i'})) = e((\widehat{g}_0)^{s_{j,i'}}, \widehat{h})^{\widehat{\gamma}_{j,i'}},$$

for uniform $\widehat{\gamma}_{j,i'} \in \mathbb{Z}^*_{\mathsf{ord}(H)}$, is uniformly distributed in a nontrivial subgroup $G'_T \subset G_T$ due to the non-degeneracy property of ENDSG. Furthermore, since A is a *weak* adversary, all the $\widehat{\mathsf{R}}_{j,n}$ are for different preimages and thus independently random. Hence, since H is a (randomly chosen) universal hash function, we have that

$$\mathsf{SD}\left((\mathsf{H}, \mathsf{H}(X))\,;\,(\mathsf{H}, U)\right) = \mathbf{O}(2^{-k}),$$

for $X \leftarrow G'_T$ and $U \leftarrow \{0,1\}^k$. A union bound yields (4.8). □

Lemma 4.2.9 (Game 4). *For any PPT adversary A in the (μ, q)-IBE-IND-CPA security experiment with IBE, it holds that*

$$\Pr[S_{A,4}] = 1/2. \tag{4.9}$$

Proof. In Game 4, for (uniform) challenge bit $b \in \{0,1\}$, we provide A with challenge ciphertexts that include a uniform k-length bitstring instead of a A-chosen b-dependent message, for each instance and challenge ciphertext. Hence, b is completely hidden from A and (4.9) follows. □

Taking (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), and (4.9) together, shows (4.1). □

From Weak (μ, q)-IBE-IND-CPA-security to (μ, q)-IBE-IND-CPA-security. Theorem 4.2.1 only considers security against weak adversaries. To achieve (μ, q)-IBE-IND-CPA-security against any PPT adversary, we have to introduce a new simple assumption (which is similar to the known BDDH assumption). Further, we need rerandomization algorithms to achieve (μ, q)-IBE-IND-CPA-security. (See below.)

A (Subgroup) Variant of the BDDH Assumption (S-BDDH). For any PPT adversary D, we have that the function

$$\begin{aligned}\mathsf{Adv}^{\text{s-bddh}}_{\mathsf{ENDSG},\mathsf{G},D}(k,n) := |\Pr\left[D(pp, \mathbf{g}, \mathbf{g}^a, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^a, \widehat{g}_0^b, \widehat{h}, \widehat{h}^b, \widehat{h}^c, e(\widehat{g}_0, \widehat{h})^{abc}) = 1\right] \\ - \Pr\left[D(pp, \mathbf{g}, \mathbf{g}^a, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^a, \widehat{g}_0^b, \widehat{h}, \widehat{h}^b, \widehat{h}^c, e(\widehat{g}_0, \widehat{h})^{z}) = 1\right]|\end{aligned}$$

is negligible in k, for $(pp, sp) \leftarrow \mathsf{SampP}(k, n)$, 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) $a, b, c, z \leftarrow \mathbb{Z}_N^*$.

Rerandomization. We use the efficient rerandomization algorithms given in [HKS15]. Essentially, [HKS15] provides efficient algorithms $\mathsf{Rerand}_\mathsf{a}$ and $\mathsf{Rerand}_\mathsf{abc}$ that rerandomize challenge tupels such that parts of the algorithms' output yield valid (rerandomized) challenge tupels. The $\mathsf{Rerand}_\mathsf{a}$-algorithm is used to rerandomize the a-exponent and the $\mathbf{T}$-element of the challenge input while $\mathsf{Rerand}_\mathsf{abc}$ rerandomizes the challenge exponents a, b, c, and the challenge $\mathbf{T}$-element. (See Proof of Lemma 4.2.10 and the rerandomization paragraph of [HKS15] for details.)

Lemma 4.2.10 (Game 3 to Game 4, (μ, q)-IBE-IND-CPA-security). *Let* G *be a group generator and* $\mathsf{Rerand}_\mathsf{abc}$, $\mathsf{Rerand}_\mathsf{a}$ *rerandomization algorithms, all as in [HKS15]. If* ENDSG *is an ENDSG system, S-BDDH holds, and* H *is a universal hash function, Game 3 and Game 4 are computationally indistinguishable. 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 is a distinguisher D with running time $t' \approx t + \mathbf{O}(\mu n k^c (q + q'))$, for some constant $c \in \mathbb{N}$, such that*

$$|\Pr[S_{A,3}] - \Pr[S_{A,4}]| \leq \mathsf{Adv}^{\text{s-bddh}}_{\mathsf{ENDSG},\mathsf{G},D}(k, 2n) + \mu q \cdot \mathbf{O}(2^{-k}). \tag{4.10}$$

Proof. In Game 3, each challenge ciphertext carries a b-dependent A-chosen

message, for $b \leftarrow \{0,1\}$, while in Game 4, each challenge ciphertext message is replace by uniform k-length b-independent bitstring.

Description. D is provided with challenge input

$$(pp, \mathbf{g}, \mathbf{g}^a, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^a, \widehat{g}_0^b, \widehat{h}, \widehat{h}^b, \widehat{h}^c, \mathbf{T}),$$

where $\mathbf{T}$ is either

$$e(\widehat{g}_0, \widehat{h})^{abc} \quad \text{or} \quad e(\widehat{g}_0, \widehat{h})^z,$$

for

$$(pp, sp) \leftarrow \mathsf{SampP}(k, 2n),$$

$$\mathbf{g} \leftarrow \mathsf{SampG}(pp),$$

$$\widehat{\mathbf{g}} = (\widehat{g}_0, \ldots, \widehat{g}_n) \leftarrow \widehat{\mathsf{SampG}}(pp, sp),$$

for $\widehat{h}$ specified in sp, for e specified in pp, and for $a, b, c, z \leftarrow \mathbb{Z}_N^*$. First, D samples $(msk_j)_j \leftarrow (H)^\mu$, sets

$$mpk_j := (pp, \mathsf{H}, m(msk_j)),$$

for all j, for $\mathsf{H} \leftarrow \mathcal{UH}$, for m specified in pp, and sends $(mpk_j)_j$ to A. Further, D defines a truly random function

$$\widehat{\mathsf{R}} : [\mu] \times \{0,1\}^n \rightarrow \langle \widehat{h} \rangle.$$

During the experiment, D answers instance-j extraction queries for $id \in \mathcal{ID}$ as

$$\overline{\mathsf{Ext}}(pp, msk_j \cdot \widehat{\mathsf{R}}(j, id), id; \mathsf{SampH}(pp)),$$

for all j. Further, A may adaptively query its Enc'-oracle; for A-chosen instance-j challenge identity $id^*_{j,i'} = id^*_{j,i',1} \ldots, id^*_{j,i',n} \in \mathcal{ID}$ and equal-length messages $(M^*_{j,i',0}, M^*_{j,i',1}) \in (\mathcal{M})^2$, for all $(j, i') \in [\mu] \times [q]$. For each fresh instance-j challenge identity $id^*_{j,i'}$ (i.e., $id^*_{j,i'}$ was not queried before by A in instance j), D computes

$$(\mathbf{g}^{a_{j,i'}}, \widehat{\mathbf{g}}^{a_{j,i'}}, \widehat{g}_0^{b_{j,i'}}, \widehat{h}^{b_{j,i'}}, \widehat{h}^{c_{j,i'}}, \mathbf{T}_{j,i'}) \leftarrow \mathsf{Rerand}_{\mathsf{abc}}(N, \mathbf{g}, \mathbf{g}^a, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^a, \widehat{g}_0^b, \widehat{h}, \widehat{h}^b, \widehat{h}^c, \mathbf{T})$$

and returns

$$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \widehat{\mathsf{R}}_{j,n}(id^*_{j,i'}), (\mathbf{g}\widehat{\mathbf{g}})^{a_{j,i'}}((g_0\widehat{g}_0)^{a_{j,i'}}, \\ (\prod_{i=1}^{n} g_{2i-id^*_{j,i',i}}\widehat{g}_{2i-id^*_{j,i',i}})^{a_{j,i'}}, \mathsf{H}(e((g_0\widehat{g}_0)^{a_{j,i'}}, msk_j)\cdot \mathbf{T}_{j,i'}) \oplus M^*_{j,i',b})$$

to A, for $b \leftarrow \{0,1\}$, for $s_{j,i'} \leftarrow \mathbb{Z}^*_N$, for all (j,i'). For a requeried challenge identity $id^*_{j,i''}$ in instance j (where $(j,i'') \in [\mu]\times[q]$ is the index of that previous query in instance j), D computes

$$(\mathbf{g}^{a'_{j,i''}}, \widehat{\mathbf{g}}^{a'_{j,i''}}, \widehat{g}_0^{b_{j,i''}}, \widehat{h}^{b_{j,i''}}, \widehat{h}^{c_{j,i''}}, \mathbf{T}'_{j,i''}) \leftarrow \\ \mathsf{Rerand}_{\mathsf{a}}(N, \mathbf{g}, \mathbf{g}^{a_{j,i''}}, \widehat{\mathbf{g}}, \widehat{\mathbf{g}}^{a_{j,i''}}, \widehat{g}_0^{b_{j,i''}}, \widehat{h}, \widehat{h}^{b_{j,i''}}, \widehat{h}^{c_{j,i''}}, \mathbf{T}_{j,i''})$$

and returns

$$\overline{\mathsf{Enc}}(pp, id^*_{j,i'}, M^*_{j,i',b}; msk_j \cdot \widehat{\mathsf{R}}_{j,n}(id^*_{j,i'}), (\mathbf{g}\widehat{\mathbf{g}})^{a_{j,i'}}((g_0\widehat{g}_0)^{a'_{j,i''}}, \\ (\prod_{i=1}^{n} g_{2i-id^*_{j,i'',i}}\widehat{g}_{2i-id^*_{j,i'',i}})^{a'_{j,i''}}, \mathsf{H}(e((g_0\widehat{g}_0)^{a'_{j,i''}}, msk_j)\cdot \mathbf{T}'_{j,i''}) \oplus M^*_{j,i'',b})$$

to A, for all (j,i''). Eventually, A outputs a guess b'. D outputs 1 if $b' = b$ and A is valid in the sense of (μ,q)-IBE-IND-CPA, else outputs 0.

Analysis. The master public keys yield the correct distribution as well as the requested user secret keys. If $\mathbf{T} = e(\widehat{g}_0, \widehat{h})^{abc}$, then the challenge ciphertext exponents (as rerandomized in $\mathsf{Rerand}_{\mathsf{abc}}$ and $\mathsf{Rerand}_{\mathsf{a}}$, respectively) are distributed $\mathbf{O}(2^{-k})$-close to the challenge ciphertext exponents in Game 3. (See rerandomization paragraph of [HKS15] for details.) For a fresh challenge identity $id^*_{j,i'}$, we have that

$$((g_0\widehat{g}_0)^{a_{j,i'}}, (\prod_{i=1}^{n} g_{2i-id^*_{j,i',i}}\widehat{g}_{2i-id^*_{j,i',i}})^{a_{j,i'}}, \\ \mathsf{H}(e((g_0\widehat{g}_0)^{a_{j,i'}}, msk_j)\cdot \mathbf{T}_{j,i'}) \oplus M^*_{j,i',b}) \\ \overset{(*)}{=} ((g_0\widehat{g}_0)^{a_{j,i'}}, (\prod_{i=1}^{n} g_{2i-id^*_{j,i',i}}\widehat{g}_{2i-id^*_{j,i',i}})^{a_{j,i'}}, \\ \mathsf{H}(e((g_0\widehat{g}_0)^{a_{j,i'}}, msk_j \cdot \widehat{h}^{b_{j,i'}c_{j,i'}})) \oplus M^*_{j,i',b}),$$

where $(*)$ holds due the orthogonality property of ENDSG. Note that we (implicitly) set $s_{j,i'} := a_{j,i'}$ and $\widehat{\gamma}_{j,i'} := b_{j,i'} \cdot c_{j,i'}$. For a requeried challenge identity $id^*_{j,i'}$, we rerandomize the previously used query value $a_{j,i'}$, for index (j, i'), and leave $\widehat{\gamma}_{j,i'}$ fixed. Otherwise, if $\mathbf{T} = e(\widehat{g}_0, \widehat{h})^z$, then the challenge ciphertext exponents are distributed $\mathbf{O}(2^{-k})$-close to the challenge ciphertext exponents in Game 4, i.e., we have that

$$\begin{aligned}
&((g_0\widehat{g}_0)^{a_{j,i'}}, (\prod_{i=1}^{n} g_{2i-id^*_{j,i',i}} \widehat{g}_{2i-id^*_{j,i',i}})^{a_{j,i'}}, \\
&\qquad\qquad \mathsf{H}(e((g_0\widehat{g}_0)^{a_{j,i'}}, msk_j) \cdot \mathbf{T}_{j,i'}) \oplus M^*_{j,i',b}) \\
&= ((g_0\widehat{g}_0)^{a_{j,i'}}, (\prod_{i=1}^{n} g_{2i-id^*_{j,i',i}} \widehat{g}_{2i-id^*_{j,i',i}})^{a_{j,i'}}, \\
&\qquad\qquad \mathsf{H}(e((g_0\widehat{g}_0)^{a_{j,i'}}, msk_j \cdot \widehat{h}^{z'_{j,i'}})) \oplus M^*_{j,i',b}),
\end{aligned}$$

for some uniform $a_{j,i'} \in \mathbb{Z}_N^*$ and $z'_{j,i'} := z_{j,i'} a_{j,i'}^{-1} \in \mathbb{Z}_N^*$ with overwhelming probability. Further, since H is a (randomly chosen) universal hash function, we have that $\mathsf{SD}\left((\mathsf{H}, \mathsf{H}(X)); (\mathsf{H}, U)\right) = \mathbf{O}(2^{-k})$, for $X \leftarrow G'_T$ and $U \leftarrow \{0,1\}^k$. Finally, via a union bound, (4.10) follows. □

Corollary 4.2.11 ((μ, q)-IBE-IND-CPA-security of IBE)**.** *Let* G *be a group generator as defined above. If* ENDSG *is an ENDSG system, S-BDDH holds, 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*

$$\begin{aligned}
\mathsf{Adv}^{(\mu,q)\text{-ibe-ind-cpa}}_{\mathsf{IBE},A}(k, n) \leq\ & \mathsf{Adv}^{\text{ls1}}_{\mathsf{ENDSG},\mathsf{G},D_1}(k, 2n) \\
&+ 2n \cdot \mathsf{Adv}^{\text{ls2}}_{\mathsf{ENDSG},\mathsf{G},D_2}(k, 2n) \\
&+ n \cdot \mathsf{Adv}^{\text{nh}}_{\mathsf{ENDSG},\mathsf{G},D_3}(k, 2n, \mu q') \\
&+ \mathsf{Adv}^{\text{s-bddh}}_{\mathsf{ENDSG},\mathsf{G},D_4}(k, 2n) + \mu q \cdot \mathbf{O}(2^{-k}), \qquad (4.11)
\end{aligned}$$

for group generator G *defined as above.*

Proof. Taking (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.10), and (4.9) together, yields (4.11). □

Chapter 5

A Generic View on Trace-and-Revoke Systems

In this section, we give a new generic view on trace-and-revoke systems which generalize new and known trace-and-revoke instantiations. (Some of the following sections and paragraphs are reproduced, partly adopted verbatim, from [HS14].)

Broadcast Encryption and Revocation Schemes. In a broadcast encryption (BE) scheme with N users, a sender is able to generate ciphertexts that only members of a "privileged" set $\mathcal{S} \subseteq \{1, \ldots, N\}$ — each given a long-lived user secret key — can decrypt correctly. There exists a large body of BE schemes under various assumptions and with various efficiency characteristics (e.g., [FN94, BGW05, Del07, BH08, GW09, FP12, LPQ12, BW13, BWZ14]). In this work, we focus on (a specific form of) revocation schemes which can be seen as variants of BE systems. But opposed to BE schemes, in revocation schemes, a set of "revoked" users $\mathcal{R} \subseteq \{1, \ldots, N\}$ is given to generate the ciphertext such that users in $\mathcal{R}$ are not eligible to decrypt the ciphertext correctly while all other users of the system are able to decrypt (after receiving a valid user secret key). Revocation schemes proposed in the literature are, e.g., [NP01, NNL01, DF03, LSW10, Wee11].

Traitor Tracing and Trace-and-Revoke Schemes. A particularly interesting property a BE system can have is traceability [CFN94], i.e., the ability to trace a "pirate" decryption box back to the corrupted user(s),

called traitor(s), who's user secret keys were used to construct it. Thus, traceability allows to identify a traitor (or a coalition of traitors). Such schemes are called traitor tracing schemes and a variety of them was proposed, e.g., [CFN94, NP98, BF99, KY02, CPP05, BSW06]. The combination of revocation and traceability is an aspiring goal. We mention that combining these properties is nontrivial (see [BW06, Section 4.1]). Nevertheless, there are schemes, e.g., [NP01, NNL01, TT01, HS02, DF02, DFKY05, BW06][1], which provide a solution to this problem. These schemes are called trace-and-revoke schemes.

Revocable Key Encapsulation Mechanisms. We focus on a specific form of revocation schemes, dubbed revocable key encapsulation mechanisms (RKEMs). An RKEM consists of four PPT algorithms: key generation, user secret key extraction, (shared key) encapsulation, and (shared key) decapsulation. Key generation outputs master public and master secret keys. The master secret key is used by the secret key extraction algorithm to derive user secret keys. On input of a revoked-user set and the master public key, the encapsulation algorithm outputs a ciphertext and a shared key. Given a user secret key and a ciphertext, decapsulation outputs a shared key or an "error message." It holds that the shared keys output by encapsulation and decapsulation, respectively, are equal if and only if the user secret key does not belong to a user in the revoked-user set that was given to encapsulate the key. A security notion for RKEMs is defined in chapter 2.

Threshold Extractable Hash Proof Systems and the Generalization of RKEMs. Wee introduces threshold extractable hash proof systems (TEHPS) [Wee11] as a generalization of extractable hash proof systems [Wee10]. Applying the concept of TEHPSs, Wee explains threshold public key encryption, threshold signatures, and revocation schemes (in detail, RKEMs), and gives TEHPS instantiations from the Decisional Diffie-Hellman (DDH), from the Computational Diffie-Hellman (CDH), and from the factoring (FAC) assumptions, which — at least in the case of factoring — led to new cryptosystems.

The Generic View of the RKEM Due to [Wee11]. We now restate

[1]Note that the schemes from [NNL01, HS02, DF02] support a different form of traitor tracing. Particularly, their main goal is to find a setting in which the pirate box is not useful anymore rather than identifying the traitor(s).

the generic RKEM of [Wee11]. The master public key consists of

$$g, g^{a_0}, g^{a_1}, \ldots, g^{a_t},$$

for a group element g and integer coefficients $a_0, a_1, \ldots, a_t$ of a polynomial $f(x) = a_0 + a_1 x + \cdots + a_t x^t$. (These public-key elements can be used to derive $g^{f(x)}$, for any integer x.) The master secret key contains the coefficients $a_0, a_1, \ldots, a_t$ and can be used to derive user secret keys via the polynomial evaluation $usk_{id} := f(id)$, for any user identity id. The ciphertext is constructed as

$$C = (\mathcal{R}, u, (u^{f(id)})_{id \in \mathcal{R}}),$$

for revoked-user set $\mathcal{R}$ and element $u = g^r$, for a random exponent r. The shared key is extracted from $s := u^{f(0)}$ via "post-processing." (Note that this scheme only allows $|\mathcal{R}| \leq t$.) Decryption works as follows: with a user secret key $usk_{id'}$ and u from the ciphertext, the value $u^{usk_{id'}} = u^{f(id')}$ is derived. If $id' \notin \mathcal{R}$, we have $t + 1$ different identities and the element $u^{f(0)} = s$ can be interpolated (via the Lagrange interpolation); s can then be post-processed to derive the shared key. (Of course, Wee proves correctness and (semantic) security.) As a consequence, depending on the domain, this yields concrete RKEMs from the factoring, the CDH, or the DDH assumption. More abstractly, Wee constructs RKEMs from TEHPSs and gives TEHPS instantiations from the above mentioned assumptions.

Our Contribution. We extend the generic view of [Wee11] by providing a TEHPS from the "Extended Decisional Diffie-Hellman" (EDDH) assumption due to [HO12]. The EDDH assumption generalizes the DDH and Decisional Composite Residuosity (DCR) assumptions. (In Figure 5.1, we give an overview of the assumptions used in this chapter.) By our first result, we obtain revocation schemes from the EDDH assumption. In particular, our generic system extends the generic view of revocation schemes from [Wee11] and, additionally, via our second result, it yields a new trace-and-revoke scheme from the DCR assumption. (This is not known for the factoring-based instance of [Wee11] and we describe why this seems to be difficult to achieve in Wee's setting.)

More on the First Result: A TEHPS Instantiation from the EDDH Assumption. By giving a slightly different generic view, we extend the work of Wee to obtain TEHPS instantiations from the extended decisional

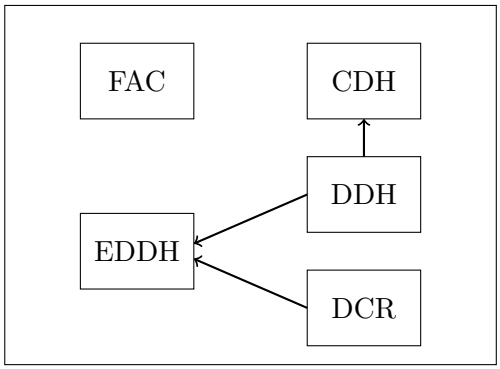

Figure 5.1: Assumptions used in this chapter and their connections. The arrows indicate that the EDDH assumption is implied by the DDH and DCR assumptions, and that the DDH assumption implies the CDH assumption

Diffie-Hellman (EDDH) assumption. Concretely, for a group $\mathcal{G}$, subset $G \subset \mathcal{G}$, subgroup $H \subseteq \mathcal{G}$, elements $g \in G$, $g^x, g,^y$ (for uniform exponents x, y), EDDH states that

$$g^{xy} \quad \text{and} \quad g^{xy} \cdot h$$

are computationally indistinguishable, for uniform $h \in H$. If $H = \mathcal{G}$, then we have the DDH assumption; if $\mathcal{G} = \mathbb{Z}^*_{N^2}$, $G = \{x^N \mid x \in \mathbb{Z}^*_N\}$, and $H = \langle 1 + N \rangle$, then we have the DCR assumption, for $N = PQ$ with distinct odd equal-length primes P, Q. In particular, our first result yields EDDH-based threshold encryption, signatures, and RKEMs. We stress that the EDDH-based instances use a potential stronger assumption (i.e., the DCR assumption) as opposed to Wee's factoring-based schemes. Nevertheless, this slightly stronger assumption enables us — via our second result — to obtain a new DCR-based trace-and-revoke scheme which, again, is not known to achieve from Wee's factoring-based RKEM.

We now turn to a high-level overview of our RKEM, which is similar to Wee's generic scheme (given above), but has ciphertexts

$$C = (\mathcal{R}, u_1, (u_1^{f(id)})_{id \in \mathcal{R}}, u_1^{f(0)} \cdot h), \tag{5.1}$$

for $u_1 \in G$ and (uniform) $h \leftarrow H$. The shared key is extracted from h. Hence, instead of directly using $u_1^{f(0)}$ to extract the shared key, we use it to blind the H-element h. The security analysis of this modified generic scheme is similar to the analysis of Wee's scheme and given below. The

only difficulties arise out of the fact that the group order of elements in G may not be known (e.g., in the case of the DCR assumption). Hence, we must avoid inversion operations in the exponent. Such inversion operations arise during Lagrange interpolation of the polynomial f in the exponent and cryptographic tools to circumvent inversion operations (e.g., "clearing the denominator") are known. (See below for more details.)

More on the Second Result: Black-box Traceability of the EDDH-based RKEM. We prove that our EDDH-based RKEM supports a "mild" form of black-box traitor tracing. That is, assuming a coalition of $T \leq (t+1)/2$ corrupted users that built any "pirate" decryption box, we are able to trace (at least) one user in that coalition. The tracing strategy only needs black-box access to the pirate box and works for imperfect boxes, where an imperfect box is allowed to decrypt well-formed ciphertexts invalidly. Further, we allow adversarially chosen revoked-user sets $\mathcal{R}$. Similar black-box tracing strategies in the revocation setting were considered in previous works, e.g., in [TT01, DFKY05]. To achieve black-box traceability in the BE setting, we note that similar techniques are common, e.g., see [BW06]. But unlike in, e.g., [TT01], our tracing algorithm works with imperfect pirate boxes that may even only work for an adversarially chosen set $\mathcal{R}$ of revoked users. The tracing model in [DFKY05] considers imperfect decryption boxes and adversarially chosen revoked users, but for a different scheme. However, we stress that our focus is on the generic view of trace-and-revoke schemes.

More Technical Details. Concerning the first result: to construct revocation schemes from the EDDH assumption — in which the group order of the elements in $G \subset \mathcal{G}$ might not be known —, we use a technique called "clearing the denominator" in the exponent. This tool was used before, but in different scenarios to ours, e.g., in [Sho00, Wee11, ABV+12], and enables us to avoid Lagrangian coefficient inversion in the exponent and to construct an EDDH-based revocation scheme.

Concerning the second result: consider "pseudo-valid" ciphertexts of the form

$$C^{\mathcal{R}}_{\mathsf{pv}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{z_{id}})_{id \in \mathcal{R}}, u_1^{f(0)} \cdot h^{z_0}) \tag{5.2}$$

for (uniform) $h \leftarrow H$ and integers $z_0, (z_{id})_{id \in \mathcal{R}}$. Under the EDDH assumption, these pseudo-valid ciphertexts from Equation 5.2 are computationally indistinguishable from the valid ciphertexts from Equation 5.1. We show

that this is even the case if one knows one user secret key. This is crucial and can be used for black-box tracing if the box contains only one user secret key. Hence, this gives rise to trace a coalition of $T = 1$ corrupted users. Basically, such tracing algorithm sends pseudo-valid ciphertexts to the pirate box and examines the output by comparing the output of the pirate box with an honestly decryption outputs using several user secret keys. (Hence, we need access to all secret keys in the system.) If the output matches, we have found a traitor. (Generally, this is the same tracing idea used in previous works, e.g., in [BW06].) As we show, this is due to the fact that the used user secret key determines the output of the pirate box. However, for $T \geq 2$, the pirate can distinguish valid and pseudo-valid ciphertexts by comparing the output under all user secret keys. (See Kiayias and Yung's work [KY01b] for a formal analysis.) We adapt our strategy by considering "semi-valid" ciphertexts of the form

$$C_{\mathsf{sv}}^{\mathcal{R},I} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{f'(id)})_{id \in \mathcal{R}}, u_1^{f(0)} \cdot h^{f'(0)}) \tag{5.3}$$

for uniform polynomial $f'(x) \in \mathbb{Z}_q[x]$ of degree $\leq t$, for G-group order q, and $f'(id) = 0$ for $id \in I$. These semi-valid ciphertexts from Equation 5.3 are indistinguishable from the valid ciphertexts from Equation 5.1 if one only knows user secret keys for user identities in I. In the tracing context, if the pirate box only contains user secret keys for identities in I, then these different forms of ciphertexts are indistinguishable. Hence, out tracing algorithm first guesses the identity set I whose user secret keys are potentially contained in the pirate box. Secondly, it checks if the semi-valid ciphertext $C_{\mathsf{sv}}^{\mathcal{R},I}$ is correctly decrypted by the pirate box. (This is essentially the "black-box confirmation" argument defined in [BF99] and used in previous works, e.g., [BW06].) It is possible that the set I contains user secret keys that are not built into the pirate box. In that case, the tracing algorithms has to ensure to output an identity whose user secret key was used to construct the pirate box. As mentioned before, similar traceability strategies were already considered, e.g., in [BF99] (but with a restriction on how the pirate box is built), and in [KY01a, BSW06, BW06] (for very different schemes). In the revocation setting the tracing technique of Tzeng and Tzeng [TT01] also considers semi-valid ciphertexts as those from Equation 5.3. However, the tracing algorithm by [TT01] assumes a pirate box with perfect decryption, and, more importantly, has to choose the revoked set $\mathcal{R}$ by itself. Dodis, Fazio, Ki-

ayias, and Yung [DFKY05] consider imperfect pirate boxes and adversarially chosen revoked users in the revocation setting, but for a different scheme. Again, we stress that the novelty of our work lies in the fact that we extend Wee's generic view of revocation schemes by providing an EDDH-based trace-and-revoke variant which, in particular, generalizes (known) DDH-based and (new) DCR-based trace-and-revoke schemes.

5.1 Preliminaries

(Binary) Relations for Hard Search Problems [Wee10, Wee11]. Following the definition of (binary) relations for hard search problems in [Wee10, Wee11], let $(R_{pp})_{pp}$ be a family of (binary) relations and let $(\mathsf{G}_{pp})_{pp}$ a family of PPT randomness extractors, both indexed by a public parameter pp. A hard-search-problem system HSP consists of two PPT algorithms:

Parameter sampling. $\mathsf{SampP}(k)$, given k, outputs public and secret parameters (pp, sp).

Relation sampling. $\mathsf{SampR}(pp)$, on input pp, outputs a (binary) relation $(u, s) \in R_{pp}$.

Further, we define one-way and indistinguishability properties for HSP as follows:

One-wayness. With overwhelming probability over all pp (that are the first output of SampP), for all u (that are the first output of SampR), there exists at most one s such that $(u, s) \in R_{pp}$.

Indistinguishability. For any PPT adversary A, the function

$$\mathsf{Adv}^{\mathsf{prg}}_{A,\mathsf{HSP},\mathsf{G}_{pp}}(k) := |\Pr[A(pp, u, \mathsf{G}_{pp}(s)) = 1] - \Pr[A(pp, u, R) = 1]|$$

is negligible in k, for all pp (that are the first output of SampP), for all $(u, s) \leftarrow \mathsf{SampR}(pp)$, for a PPT randomness extractor G_{pp}, and for uniformly sampled random element R in the co-domain of G_{pp}.

Threshold Extractable Hash Proof Systems. We first restate the definition of threshold extractable hash proof systems (TEHPSs) from [Wee11].

Wee explains several cryptosystems, i.e., threshold encryption, threshold signatures, and RKEMs as arising from TEHPSs for hard search problems with instances u and solution s (defined as above). We define a family of hash functions $(\mathsf{H}_{hk})_{hk}$ that are indexed by a public (hash) key hk. H_{hk} takes as input a tag t (from a tag space $\mathcal{T}$) and an instance u, and outputs a hash value $\mathsf{H}_{hk}(t,u)$. Further, let $\mathsf{HSP} = (\mathsf{SampP}, \mathsf{SampR})$ be a hard-search-problem system. A TEHPS TEHPS with tag space $\mathcal{T}$ consists of the PPT algorithms $(\mathsf{Gen}, \mathsf{Ext}, \mathsf{Pub}, \mathsf{Priv}, \mathsf{TExt})$ as follows:

Key generation. $\mathsf{Gen}((pp, sp), k, t)$, given public and secret parameters

$$(pp, sp) \leftarrow \mathsf{SampP}(k),$$

security parameter k, and threshold parameter $t \in \mathbb{N}$, generates a master public (hash) and master secret keys (hk, msk).

Secret-key extraction. $\mathsf{Ext}(msk, t)$, given the master secret key msk and a tag $t \in \mathcal{T}$, generates a user secret key usk_t for the tag t.

Public evaluation. $\mathsf{Pub}(hk, t, r)$, given a public key hk, a tag $t \in \mathcal{T}$, and random value r, outputs a hash value

$$\mathsf{H}_{hk}(t,u),$$

for $(u, s) = \mathsf{SampR}(pp; r)$.

Private evaluation. $\mathsf{Priv}(usk_t, u)$, given a user secret key usk_t and an instance u, outputs a hash value

$$\mathsf{H}_{hk}(t,u).$$

Threshold Extraction. $\mathsf{TExt}(u, (t_i, \tau_i)_{i=1}^{t+1})$, given an instance u, $t+1$ tags $(t_i)_{i=1}^{t+1} \in (\mathcal{T})^{t+1}$, and $t+1$ hash values $(h_i)_{i=1}^{t+1}$, outputs a value s or $\{\perp\}$.

For all $k, t \in \mathbb{N}$, with overwhelming probability over all $(pp, sp) \leftarrow \mathsf{SampP}(k)$, for all $(hk, msk) \leftarrow \mathsf{Gen}((pp, sp), k, t)$, for all random values r, for all $(u, s) \leftarrow \mathsf{SampR}(pp; r)$, we define correctness, $(t+1)$-extraction, and t-simulation of a TEHPS as follows:

Correctness. For all $t \in \mathcal{T}$, for all $usk_t \leftarrow \mathsf{Ext}(msk, t)$, we have that

$$\mathsf{Pub}(hk, t, r) = \mathsf{H}_{hk}(t, u) = \mathsf{Priv}(usk_t, u).$$

$(t+1)$**-extraction.** For all distinct tags $(t_i)_{i=1}^{t} \in (\mathcal{T})^{t+1}$, and for all hash values $(h_i := \mathsf{H}_{hk}(t_i, u))_i$, for $s' = \mathsf{TExt}(u, (t_i, h_i)_i)$, we have that $(u, s') \in R_{pp}$.

t**-simulation.** For all distinct tags $(t_i)_{i=1}^{t} \in (\mathcal{T})^{t}$, there exists a PPT algorithm Gen' such that distributions of

$$\omega = (hk, usk_{t_1}, \ldots, usk_{t_t}) \quad \text{and} \quad \omega' = (hk,' usk'_{t_1}, \ldots, usk'_{t_t})$$

are statistically close. Concretely, we have that

$$\begin{aligned} &\{\omega : (hk, msk) \leftarrow \mathsf{Gen}((pp, sp), k, t), (usk_{t_i} \leftarrow \mathsf{Ext}(msk, t_i))_i\} \\ &\overset{s}{\approx} \{\omega' : (hk', usk'_{t_1}, \ldots, usk'_{t_t}) \leftarrow \mathsf{Gen}'(pp, t_1, \ldots, t_t)\}, \end{aligned}$$

where $\overset{s}{\approx}$ denotes statistically indistinguishable.

5.2 An EDDH-based TEHPS Instance

In this section, we construct a new EDDH-based threshold extractable hash proof system. As opposed to the DDH-based construction in [Wee11], where the orders of each is group is known, the orders of elements in a subset $G \subset G$, given in the EDDH-assumption, may not be known. Hence, we must avoid inversion operations in the exponent. To this end, we use a technique called "clearing the denominator" that in a similar way was used before, but in different scenarios, e.g., in [Sho00, Wee11, ABV+12].

We specify a (binary) relation for the EDDH problem as

$$R_{pp} = \{(u, s) \in ((G \times \mathcal{G}) \times H) \mid u = (u_1, u_2), u_2 = u_1^{sp} s\},$$

for $g \leftarrow G$, for $s \leftarrow H$, and for $sp \leftarrow \mathcal{K}$, where $\mathcal{G}$, G, H, and $\mathcal{K}$ are determined by SampP (given below). We instantiate the HSP-algorithms SampP and SampR as follows:

– $\mathsf{SampP}(k)$, given k, fixes an abelian group $\mathcal{G}$ with order q and an (efficiently samplable) subgroup $H \subseteq \mathcal{G}$ of order n. (We assume that a (proper) lower bound d on the smallest prime divisor of n is known.) Further, SampP sets $G \subset \mathcal{G}$ to be efficiently samplable and $\mathcal{K} := [B]$, for $B := B' \cdot 2^k$, for an upper bound $B' \geq q$, where B' is determined by SampP. (We need that for $x \leftarrow \mathcal{K}$, the value x mod q is statistically close to uniform.) SampP then samples $g \leftarrow G$, $sp \leftarrow \mathcal{K}$ and outputs public and secret parameters as

$$pp := (n, g, g^{sp}) \quad \text{and} \quad sp.$$

– $\mathsf{SampR}(pp; r)$, given pp and random value $r \in \mathcal{K}$, samples $s \leftarrow H$ and outputs

$$(u, s) := ((g^r, (g^{sp})^r \cdot s), s).$$

Concerning the randomness extractor, we set $\mathsf{G}_{pp}(s) := \mathsf{G}(s)$ (for G from the EDDH assumption). Now, we are able to construct:

An EDDH-based Threshold Extractable Hash Proof System. We construct an EDDH-based TEHPS $\mathsf{TEHPS} = (\mathsf{Gen}, \mathsf{Ext}, \mathsf{Pub}, \mathsf{TExt}, \mathsf{Priv})$ with tag space $\mathcal{T} := [\min\{d, B\}] \subset \mathbb{Z}$, with d and B as above, from a EDDH-based HSP system $\mathsf{HSP} = (\mathsf{SampP}, \mathsf{SampR})$ as above. To this end, we fix a hash function $\mathsf{H}_{hk}(t, u) := u_1^{f(t)}$, for some tag $t \in \mathcal{T}$, for f specified during key generation below, for $(u = (u_1, u_2), s) \leftarrow \mathsf{SampR}(pp; r)$, for randomness $r \in \mathcal{K}$, and for pp as the first output of $\mathsf{SampP}(k)$.

Key generation. $\mathsf{Gen}((pp, sp), k, t)$, given k, t, and parameters $(pp, sp) \leftarrow \mathsf{SampP}(k)$, for $pp =: (n, g, g^{sp})$, chooses a polynomial

$$f(x) := sp + a_1 x + \cdots + a_t x^t$$

over $\mathcal{K}$, with uniform exponents $(a_i)_{i=1}^t$. The output is the master public key

$$hk := (n, \widetilde{g}, \widetilde{g}^{sp}, (\widetilde{g}^{a_i})_{i=1}^t),$$

with $\widetilde{g} := g^v$, for uniform $v \leftarrow \mathcal{K}$, and master secret key

$$msk := (sp, (a_i)_{i=1}^t).$$

Secret-key extraction. $\mathsf{Ext}(msk, t)$, given msk and $t \in \mathcal{T}$, returns $usk_t := f(t)$.

Public evaluation. $\mathsf{Pub}(hk, t, r)$, given hk, tag $t \in \mathcal{T}$, and randomness $r \in \mathcal{K}$, computes

$$\big(\widetilde{g}^{sp} \cdot \prod_{i=1}^{t} (\widetilde{g}^{a_i})^{t^i}\big)^r \qquad \Big(= \big(\widetilde{g}^{f(t)}\big)^r = u_1^{f(t)} = \mathsf{H}_{hk}(t,u)\Big),$$

with $(u, s) = \mathsf{SampR}(k, (n, \tilde{g}, \tilde{g}^{sp}); r)$.

Private evaluation. $\mathsf{Priv}(usk_t, u)$, given usk_t and $u =: (u_1, u_2)$, outputs

$$u_1^{usk_t} \qquad \Big(= u_1^{f(t)} = \mathsf{H}_{hk}(t,u)\Big).$$

Threshold extraction. $\mathsf{TExt}(u, (t_i, h_{t_i})_{i=1}^{t+1})$, given $u = (u_1, u_2)$, tags $(t_i)_i \in (\mathcal{T})^{t+1}$, and hash values $(h_{t_i})_i$, computes fractional Lagrangian coefficients

$$L_i(0) = \prod_{j=1, i\neq j}^{t+1} \frac{-t_j}{t_i - t_j}$$

such that

$$f(0) = \sum_{i=1}^{t+1} L_i(0) \cdot f(t_i) \bmod q.$$

(The Lagrangian coefficients can be computed if all tags $(t_i)_i$ are distinct. If the tags are not distinct, then we output $\{\perp\}$.) For

$$\Delta := \mathrm{lcm}\{ \prod_{i,j\in[t+1], i\neq j} (t_i - t_j) \in \mathbb{Z}\},$$

the values $\Delta \cdot L_i(0)$, for all $i \in [t+1]$, are integers. Hence, we are able to extract and output the value

$$\big(\big(\prod_{i=1}^{t+1} h_{t_i}^{\Delta L_i(0)}\big)^{-1} \cdot u_2^{\Delta}\big)^{\Delta^{-1} \bmod n}.$$

We now show correctness, $(t+1)$-extraction, and t-simulation of the constructed EDDH-based TEHPS.

Claim 5.2.1. *For all $t \in \mathbb{N}$, TEHPS from above is correct, $(t+1)$-extractable, and t-simulatable.*

Proof. For all $k, t \in \mathbb{N}$, with overwhelming probability over $(pp, sp) \leftarrow \mathsf{SampP}(k)$, for all r, for all $(u, s) \leftarrow \mathsf{SampR}((n, \tilde{g}, \tilde{g}^{sp}); r)$, with $u = (u_1, u_2)$, for all $(hk, msk) \leftarrow \mathsf{Setup}((pp, sp), t)$, for all tags $t \in \mathcal{T}$, and all $usk_t \leftarrow \mathsf{Ext}(msk, t)$, we have:

Correctness. Correctness is easy to verify, i.e., it holds that

$$\mathsf{Pub}(hk, t, r) = \mathsf{H}_{hk}(t, u) = \mathsf{Priv}(usk_{t_i}, u).$$

$(t+1)$-extraction. For all distinct tags $(t_i)_{i=1}^{t+1} \in (\mathcal{T})^{t+1}$, all hash values $(h_i := \mathsf{H}_{hk}(t_i, u))_{i=1}^{t+1} (= (u_1^{f(t_i)})_i)$, for Δ and fractional Lagrangian coefficients $L_i(0)$ as above, $\mathsf{TExt}(u, (t_i, h_{t_i})_i)$ yields

$$\begin{aligned}
&\Big(\Big(\prod_{i=1}^{t+1} h_{t_i}^{\Delta L_i(0)}\Big)^{-1} \cdot u_2^{\Delta}\Big)^{\Delta^{-1} \bmod n} \overset{(*)}{=} \Big(\big(u_1^{\Delta f(0)}\big)^{-1} \cdot (u_1^{sp} \cdot s)^{\Delta}\Big)^{\Delta^{-1} \bmod n} \\
&= \big(u_1^{-\Delta sp} \cdot u_1^{\Delta sp} \cdot s^{\Delta}\big)^{\Delta^{-1} \bmod n} = s.
\end{aligned}$$

Recall that all $\Delta \cdot L_i(0)$, for $i \in [t+1]$, are integers and that Lagrangian interpolation in the exponent is used in $(*)$. Hence, we obtain s such that $(u, s) \in R_{pp}$.

t-simulation. We give a PPT algorithm Gen'. For all distinct tags $(t_i)_{i=1}^{t+1} \in (\mathcal{T})^{t+1}$, Gen' chooses uniformly $y_1, \ldots, y_t \leftarrow \mathcal{K}$ and sets $f(t_i) := y_i$, for $i \in [t]$. Further, it sets $\widehat{g} := g^v$, for uniform $v \leftarrow \mathcal{K}$, and $\widehat{g}^{f(0)} := (g^{sp})^v = \widehat{g}^{sp}$. Note, that this will uniquely define a polynomial f of degree $\leq t$. Let Δ be as above, but with $t_{t+1} = 0$. That (implicitly) determines a vector

$$(\Delta a_0, \Delta a_1, \ldots, \Delta a_t)^{\top} := (\Delta \cdot V^{-1}_{t_{t+1}, t_1, \ldots, t_t}) \cdot (sp, y_1, \ldots, y_t)^{\top}.$$

(That is, every Δa_i can be written as linear combination of the y_i, with appropriate integer coefficients. Here, again, we use Δ to "clear the denominator" of V^{-1}'s entries.) Subsequently, for $\widetilde{g} := \widehat{g}^{\Delta}$, Gen' outputs

$$(n, \widetilde{g}, \widetilde{g}^{a_0}, \widetilde{g}^{a_1}, \ldots, \widetilde{g}^{a_t}) \quad \text{and} \quad (usk_{t_1}, \ldots, usk_{t_t}) := (y_1, \ldots, y_t).$$

Hence, the distribution of the output of Gen' and the distribution of

$$(hk, (\mathsf{Ext}(msk, t_i))_{i=1}^{t})$$

are statistically indistinguishable. □

Now, by [Wee11, Theorems 3], we derive non-adaptively t-RKEM-IND-CPA-secure RKEMs from the hardness of the EDDH assumption which yields new DCR-based RKEMs.

5.3 Traceability of an EDDH-based RKEM

We start by recapping how to build RKEMs from TEHPSs (from [Wee11]). Further, the following paragraphs, subsections, theorems, lemmas, claims, and proofs are restated, partly adopted verbatim, from [HS14].

RKEMs from TEHPSs. Following [Wee11], we recap the construction of an RKEM $\mathsf{RKEM} = (\mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ with identity space $\mathcal{ID} := \mathcal{T}$ from a TEHPS $\mathsf{TEHPS} = (\mathsf{TEHPS.Gen}, \mathsf{TEHPS.Ext}, \mathsf{Pub}, \mathsf{TExt}, \mathsf{Priv})$ with tag space $\mathcal{T}$ and an HSP system $\mathsf{HSP} = (\mathsf{SampP}, \mathsf{SampR})$ as follows:

Key generation. $\mathsf{Gen}(k, t)$, given k and revocation threshold $t \in \mathbb{N}$, samples $(pp, sp) \leftarrow \mathsf{SampP}(k)$ and outputs master public and master secret keys

$$(mpk, msk) := \mathsf{TEHPS.Gen}((pp, sp), t).$$

Secret-key extraction. $\mathsf{Ext}(msk, id)$, given msk and $id \in \mathcal{ID}$, returns

$$usk_{id} \leftarrow \mathsf{TEHPS.Ext}(msk, id).$$

Encapsulation. $\mathsf{Enc}(mpk, \mathcal{R})$, given master public key mpk and $\mathcal{R} \subseteq \mathcal{ID}$ of size exactly t, chooses a random value r, samples

$$(u, s) \leftarrow \mathsf{SampR}(mpk; r),$$

and computes $h_{id} := \mathsf{Pub}(hk, id, r)$, for $id \in \mathcal{R}$. The ciphertext is given by

$$C := (\mathcal{R}, u, (h_{id})_{id \in \mathcal{R}}),$$

the key is $K := \mathsf{G}_{pp}(s)$.

Decapsulation. $\mathsf{Dec}(id, usk_{id}, C)$, given usk_{id} and C, retrieves

$$s := \mathsf{TExt}(u, \mathcal{R} \cup \{id\}, (h_{id})_{id \in \mathcal{R}}, \mathsf{Priv}(usk_{id}, u))$$

and outputs $K := \mathsf{G}_{pp}(s)$.

Correctness is easy to verify. For t-RKEM-IND-CPA-security, we point to [Wee11, Theorem 3]. Hence, as a result, we derive EDDH-based RKEMs.

Trace-and-Revoke Schemes. Essentially, a trace-and-revoke system is the connection of a revocation scheme with a tracing algorithm. As mentioned before, combining these is nontrivial (see [BW06, Section 4.1]). Following the tracing definitions in [BF99, DFKY05, BW06], we define traceability of an RKEM analogously. Intuitively, we require an efficient algorithm Trace that can, from oracle access to a stateless pirated box $\mathcal{B}$, deduce the identity of at least one party that has been involved in the construction of $\mathcal{B}$. More concretely, suppose an adversary A corrupts a number of devices (i.e., obtains a number of user keys usk_{id}), and constructs a pirate box $\mathcal{B}$. Suppose that $\mathcal{B}$ successfully decrypts ciphertexts for an adversarially specified set $\mathcal{R}$ of revoked users. Then, we want that Trace, given oracle access to $\mathcal{B}$, can deduce at least one of the identities id whose device A has corrupted. We will also define a relaxation of traceability, dubbed sid-traceability, in which the adversary has to commit to corrupted identities in advance, before even seeing the public key.

(Sid-)traceable RKEMs. We say that that an adversary A is T-valid if, in experiment $\mathsf{Exp}^{\mathsf{trace}}_{\mathsf{RKEM},\mathsf{Trace},A}$ (defined in Figure 5.2), it always chooses $t \geq T$, it always outputs a set $\mathcal{R}$ of size at most t, and it always makes at most T Ext queries. (Note that this definition does not actually depend on Trace, and that t is specified by A itself.) Furthermore, for given $pk, \mathcal{R}$, we define the quality of a pirate box $\mathcal{B}$ output by A as

$$\mathrm{Q}_{\mathcal{B},\mathcal{R}} := \Pr\left[\mathcal{B}(C) = K \mid (C, K) \leftarrow \mathsf{Enc}(pk, \mathcal{R})\right].$$

An RKEM RKEM is (T, ε)-traceable if there exists a PPT algorithm Trace (that may depend on T and ε), so that for every PPT T-valid A,

$$\mathsf{Adv}^{\mathsf{trace}}_{\mathsf{RKEM},A}(k) := \Pr\left[\mathsf{Exp}^{\mathsf{trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}(k) = 1\right]$$

is negligible. RKEM is (T, ε)-traceable under selective-identity attacks (short: (T, ε)-sid-traceable) if the analogous statement holds with respect to

$$\mathsf{Adv}^{\text{sid-trace}}_{\mathsf{RKEM},A}(k) := \Pr\left[\mathsf{Exp}^{\text{sid-trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}(k) = 1\right]$$

and $\mathsf{Exp}^{\text{sid-trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}$, defined in Figure 5.2, in which A has to output an identity set $\mathcal{C}$ of corrupted users of size at most t in advance.

Experiment $\mathsf{Exp}^{\text{trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}(k)$
$t \leftarrow A(k)$
$(mpk, msk) \leftarrow \mathsf{Gen}(k, t)$
$(\mathcal{B}, \mathcal{R}) \leftarrow A^{\mathsf{Ext}(msk,\cdot)}(mpk)$
$id \leftarrow \mathsf{Trace}^{\mathcal{B}(\cdot)}(msk, \mathcal{R})$
if A has queried $\mathsf{Ext}(msk, id)$
or $\mathrm{Q}_{\mathcal{B},\mathcal{R}} < \varepsilon$ return 0
return 1

Experiment $\mathsf{Exp}^{\text{sid-trace}}_{\mathsf{RKEM},\mathsf{Trace},A,\varepsilon}(k)$
$(t, \mathcal{C}) \leftarrow A(k)$
$(mpk, msk) \leftarrow \mathsf{Gen}(k, t)$
$\forall id \in \mathcal{C}$: $usk_{id} \leftarrow \mathsf{Ext}(msk, id)$
$(\mathcal{B}, \mathcal{R}) \leftarrow A(pk, (usk_{id})_{id\in\mathcal{C}})$
$id \leftarrow \mathsf{Trace}^{\mathcal{B}(\cdot)}(msk, \mathcal{R})$
if $id \in \mathcal{C}$ or $\mathrm{Q}_{\mathcal{B},\mathcal{R}} < \varepsilon$ return 0
return 1

Figure 5.2: Security experiments for (sid-)traceability of an RKEM.

From Sid-traceability to Traceability. There is a trivial (yet expensive) way to convert sid-traceable RKEMs into traceable ones. Namely, we can simply guess the identities for which an adversary (adaptively) requests user keys. Concretely:

Lemma 5.3.1. *Let* RKEM *by a (T, ε)-sid-traceable RKEM with N identities. If $\binom{N}{T}$ is polynomial in k, then* RKEM *is also (T, ε)-traceable (with the same* Trace *algorithm). Concretely, for every adversary A on* RKEM*'s traceability,*

there is an adversary A' of roughly the same complexity on RKEM*'s sid-traceability, such that*

$$\mathsf{Adv}^{\text{sid-trace}}_{\mathsf{RKEM},A'}(k) \geq \mathsf{Adv}^{\text{trace}}_{\mathsf{RKEM},A}(k)/\binom{N}{T}.$$

Proof. First, A' outputs a uniformly chosen subset $\mathcal{C} \subseteq \mathcal{ID}$ of size T, and receives a public key pk along with user keys usk_{id} for $id \in \mathcal{C}$. Then A' internally simulates A, answering A's Ext queries using the usk_{id}. If A requests a user secret key for an identity $id \notin \mathcal{C}$, then A' fails. Otherwise, A' relays A's output $(\mathcal{B}, \mathcal{R})$. Since A' chooses $\mathcal{C}$ independently, the event that A' fails is independent of A's output. Besides, the probability that A' does not fail is at least $1/\binom{N}{T}$, which is significant. □

Relation to the Traceability of an EDDH-based RKEM. Our second result (below) shows the $((t+1)/2, \varepsilon)$-sid-traceability of an EDDH-based RKEM based on threshold extractable hash proofs. Our corresponding tracing algorithm will have a runtime that is linear in $\binom{N}{T}$. Thus, in that case, $\binom{N}{T}$ must be polynomial anyway, and the loss in Lemma 5.3.1 seems acceptable.

More on Our Tracing Strategy. We propose a tracing strategy that is similar to the tracing techniques in the revocation setting given by [TT01, DFKY05]. (In the BE setting, similar tracing techniques are also known, e.g., [BW06].) However, we stress that the tracing algorithm of [TT01] assumes a pirate box with perfect decryption, i.e., $\varepsilon = 1$, and chooses the revoked set $\mathcal{R}$ by itself. The tracing mode in [DFKY05] also considers imperfect decryption boxes, adversarially chosen revoked user sets, and, additionally, allows of querying user secret keys adaptively. (This is possible since their scheme allows to change the public key continuously even after the system setup.) Additionally, both, i.e., [TT01, DFKY05], only address the DDH setting. Nevertheless, we stress that the novelty of our work lies in the fact that we propose a new generic view of trace-and-revoke schemes.

5.3.1 $(1, 2/3)$-Sid-traceability of RKEM

We are now ready to state our second result; i.e., we show the traceability of RKEM which is an EDDH-based RKEM as defined and constructed in

Section 5.2. (This immediately translates to an EDDH-based trace-and-revoke scheme.) As a warmup, we first showcase the $(1, 2/3)$-sid-traceability of RKEM.

Informal Proof Strategy. To explain the overall idea of our tracing algorithm, observe that the decryption of a ciphertext generated by Enc does not depend on which user key was used to decrypt. (This is necessary for correctness.) Hence, we cannot expect that a pirate box $\mathcal{B}$ can be traced by feeding it valid ciphertexts generated by Enc. Instead, we will feed $\mathcal{B}$ random ciphertexts of the form

$$C^{\mathcal{R}}_{\mathsf{pv}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{z_{id}})_{id \in \mathcal{R}}, u_1^{f(0)} \cdot h^{z_0}) \tag{5.4}$$

for uniform $h \in H$ and z_{id}, z_0. We will show that for such random ciphertexts, the result of the (honest) decryption depends on the identity of the used user key usk_{id}. Furthermore, a suitable reduction to the EDDH assumption will show that honestly generated ciphertexts are indistinguishable from random ones. Hence, Trace can go through the set of all possible identities id, and check how often $\mathcal{B}(C^{\mathcal{R}}_{\mathsf{pv}})$ coincides with $\mathsf{Dec}(id, usk_{id}, C^{\mathcal{R}}_{\mathsf{pv}})$. In case $\mathcal{B}$ outputs the same as Dec with probability close to $2/3$, chances are that we have found the pirate identity. We can formalize these claims:

Theorem 5.3.2 ($(1, 2/3)$-sid-traceability of RKEM)**.** *Assuming the EDDH assumption, we have that the RKEM* $\mathsf{RKEM} = (\mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$*, with identity space* $\mathcal{ID}$*, polynomial number* N *of identities, and key derivation function* $\mathsf{G}(s) = s$*, is* $(1, 2/3)$*-sid-traceable. The corresponding tracing algorithm* Trace *runs for* $\mathbf{O}(kN \log N)$ *steps, and makes* $\mathbf{O}(k \log N)$ *oracle queries. Concretely, for every* T*-valid adversary* A*, there is an EDDH adversary* D*, such that*

$$\left|\mathsf{Adv}^{\mathsf{trace}}_{\mathsf{RKEM},A}(k)\right| \leq \mathbf{O}(2^{-k}),$$

for all k *that satisfy*

$$\left|\mathsf{Adv}^{\mathsf{eddh}}_{G,H,D}(k)\right| \leq 1/9 - \varepsilon_{\mathsf{G}},$$

for negligible ε_{G}*.*

Proof. **The tracing algorithm.** First, $\mathsf{Trace}^{\mathcal{B}(\cdot)}(msk, \mathcal{R})$ approximates for every identity $id \in \mathcal{ID}$ the *random quality*

$$\mathrm{RQ}^{id}_{\mathcal{B},\mathcal{R}} := \Pr\left[\mathcal{B}(C^{\mathcal{R}}_{\mathsf{pv}}) = \mathsf{Dec}(id, usk_{id}, C^{\mathcal{R}}_{\mathsf{pv}})\right],$$

where the probability is over $\mathcal{B}$'s random coins and random $C_{\mathsf{pv}}^{\mathcal{R}}$ as in (5.4). Concretely, say that for each $id \notin \mathcal{R}$, we check

$$\mathcal{B}(C_{\mathsf{pv}}^{\mathcal{R}}) = \mathsf{Dec}(id, usk_{id}, C_{\mathsf{pv}}^{\mathcal{R}})$$

for $\mathbf{O}(k \cdot \log N)$ independent values of $C_{\mathsf{pv}}^{\mathcal{R}}$. Then a standard argument (i.e., Hoeffding's inequality and a union bound) shows that we obtain approximations $\widetilde{\mathrm{RQ}}_{\mathcal{B},\mathcal{R}}^{id}$ of $\mathrm{RQ}_{\mathcal{B},\mathcal{R}}^{id}$, such that

$$\text{for all } id\text{:} \quad \left|\widetilde{\mathrm{RQ}}_{\mathcal{B},\mathcal{R}}^{id} - \mathrm{RQ}_{\mathcal{B},\mathcal{R}}^{id}\right| < 1/9, \tag{5.5}$$

except with probability $\mathbf{O}(2^{-k})$. After having obtained all these $\widetilde{\mathrm{RQ}}_{\mathcal{B},\mathcal{R}}^{id}$, Trace outputs an identity with maximal $\widetilde{\mathrm{RQ}}_{\mathcal{B},\mathcal{R}}^{id}$. The whole process takes $\mathbf{O}(Nk \log N)$ steps and (if we re-use $\mathcal{B}$-queries across different identities) $\mathbf{O}(k \log N)$ $\mathcal{B}$-queries.

Why tracing works. To analyze Trace, consider an adversary A in the 1-sid-traceability experiment. We assume without loss of generality that A always requests exactly one user key. Let id^* be the corresponding identity. Furthermore, we assume that the set $\mathcal{R}$ that A finally outputs contains exactly t identities, which we denote by $id_1^*, \ldots, id_t^*$. We finally assume $id^* \notin \mathcal{R}$. (If $id^* \in \mathcal{R}$, then any pirate box $\mathcal{B}$ that is able to decrypt with significant probability would contradict RKEM's semantic security.) We denote by $\mathcal{B}$ the pirate box that A eventually outputs.

Claim 5.3.3. *There is a EDDH distinguisher D whose runtime is essentially that of the sid-traceability experiment with A, such that*

$$\mathrm{Q}_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}_{\mathcal{B},\mathcal{R}}^{id^*} = \mathsf{Adv}_{G,H,D}^{\mathsf{eddh}}(k). \tag{5.6}$$

Proof. On challenge input

$$(n = \mathsf{ord}(H), g, u_1, g^y, Z = u_1^y h^b),$$

where either $b = 0$ or $b = 1$, D runs the first stage of the sid-traceability experiment to obtain t and $\mathcal{C} = \{id^*\}$ from A. It then constructs an RKEM

public key as follows. First, D re-randomizes its input to obtain t tuples

$$(g^{y_1}, Z_1 := u_1^{y_1} h^{bz_1}), \ldots, (g^{y_t}, Z_t := u_1^{y_t} h^{bz_t})$$

with

$$g^{y_i} := (g^y)^{\alpha_i} g^{\beta_i} \quad \text{and} \quad Z_i := Z^{\alpha_i} u_1^{\beta_i} = u_1^{y\alpha_i} h^{b\alpha_i} u_1^{\beta_i} = u_1^{y_i} h^{b\alpha_i},$$

for $i \in [t]$ and exponents α_i, β_i that are (statistically close to) uniform modulo n and modulo q. Hence, the y_i and $z_i := \alpha_i \bmod n$, for all i, are independently uniform. Now, choose an arbitrary set $\{id_1, \ldots, id_t\} \subset \mathcal{T}$ of t distinct identities that does not contain id^* and sample $y^* \leftarrow \mathcal{K}$. We (implicitly) define $f(x) := a_0 + a_1 x + \cdots + a_t x^t$ as the unique $\leq t$-degree polynomial over $\mathcal{K}$ that satisfies $f(id_i) = y_i$, for $i \in [t]$, and $f(id^*) = y^*$. Note that D cannot directly compute f. However, D does know id^* and all id_i, as well as all $\widehat{g}^{y_i} = \widehat{g}^{f(id_i)}$ and $\widehat{g}^{y^*} = \widehat{g}^{f(id^*)}$, with $\widehat{g} := g^v$, for uniform exponent v). Hence, for $\Delta := \mathrm{lcm}\{\prod_{i,j \in [t+1], i \neq j} (id_i - id_j) \in \mathbb{Z}\}$, with $id_{t+1} := id^*$, D can compute

$$(\widehat{g}^{\Delta a_0}, \ldots, \widehat{g}^{\Delta a_t})^\top := \left(\Delta \cdot V^{-1}_{id_1, \ldots, id_t, id^*}\right) \circ (\widehat{g}^{y_1}, \ldots, \widehat{g}^{y_t}, \widehat{g}^{y^*})^\top$$

without modular inversion in the exponent. Thus, for $\widetilde{g} := \widehat{g}^{\Delta}$, D can set up a public key $pk := (n, \widetilde{g}, (\widetilde{g}^{a_i})_{i=0}^t)$ for A, and run the next stage of the 1-sid-traceability experiment (using y^* as a user key for identity id^*). Now, D obtains a set $\mathcal{R} = \{id_1^*, \ldots, id_t^*\}$ of t revoked identities and a pirate box $\mathcal{B}$ from A. Consider the following $(t+1) \times (t+1)$-matrix $M = (M_{i,j})$ over $\mathcal{K}$ given by

$$M := V_{(id_1^*, \ldots, id_t^*, id^*)} \cdot V^{-1}_{(id_1, \ldots, id_t, id^*)}, \text{ so that } M \cdot \begin{pmatrix} f(id_1) \\ \vdots \\ f(id_t) \\ f(id^*) \end{pmatrix} = \begin{pmatrix} f(id_1^*) \\ \vdots \\ f(id_t^*) \\ f(id^*) \end{pmatrix}. \quad (5.7)$$

Note that M only depends on (and can be computed efficiently from) the id_i, the id_i^*, and id^*. Furthermore, since all respective identities in $\{id_i\}_i \cup \{id^*\}$ and $\{id_i^*\}_i \cup \{id^*\}$ are distinct, M is invertible. Now, D computes the vector

$$((Z_1')^\Delta, \ldots, (Z_t')^\Delta, (Z_{t+1}')^\Delta)^\top := (\Delta \cdot M) \circ (Z_1, \ldots, Z_t, Z_{t+1})^\top, \quad (5.8)$$

with $Z_{t+1} := u_1^{y^*}$. With these $t+1$ values $(Z_i')^\Delta$ and $t+1$ identities in $\{id_i^*\}_i \cup \{id^*\}$, we are able to obtain

$$(Z_0')^\Delta := (u_1^{f(0)} h^{f'(0)\cdot b})^\Delta$$

through Lagrangian interpolation (without modular inversion in the exponent), with implicitly defined $\leq t$-degree polynomial f' such that $f'(id_i^*) = z_i$, for all i, and $f'(id^*) = 0$. Intuitively, f' is the "h-exponent" of the Z_i, resp. Z_i'. Finally, D hands a ciphertext

$$C := (\mathcal{R}, u_1, (Z_1')^\Delta, \ldots, (Z_t')^\Delta, (Z_0')^\Delta \cdot s),$$

with $(Z_0')^\Delta$ as above and $(Z_i')^\Delta$ as in (5.8, for $i \in [t]$, and uniform $s \in H$) to $\mathcal{B}$ to obtain a potential decryption K. If $K = \mathsf{Dec}(id^*, y^*, C)$, then D outputs 1, else 0. This completes our description of D.

First observe that when $b = 0$, $Z_i = u_1^{y_i} = u_1^{f(id_i)}$ for all i, then (5.7) implies $Z_i' = u_1^{f(id_i^*)}$ for all i. Hence, C is distributed exactly like an honest encryption $\mathsf{Enc}(pk, \mathcal{R})$, and by correctness of RKEM, we have

$$\begin{aligned} &\Pr\left[D(1^k, n, g, u_1, g^y, Z) = 1 \mid Z = u_1^y\right] \\ =\ &\Pr\left[\mathcal{B}(C) = K \mid (C, K) \leftarrow \mathsf{Enc}(pk, \mathcal{R})\right] \\ =\ &\mathrm{Q}_{\mathcal{B},\mathcal{R}}, \end{aligned} \tag{5.9}$$

Conversely, assume $b = 1$, we have $Z_i = u_1^{y_i} h^{z_i}$, for all $i \in [t]$, and $Z_{t+1} := u_1^{y^*}$. Consider the (implicitly) defined degree-$\leq t$ polynomial f' with $f'(id_i) = z_i$ for $i \in [t]$ and $f'(id^*) = 0$. In other words, $Z_i = u_1^{y_i} h^{f'(id_i)}$. By the interpolation properties of M, this sets $Z_i' = u_1^{y_i} h^{f'(id_i^*)}$ and thus $Z_0' = u_1^{f(0)} h^{f'(0)}$. The ciphertext now includes t values $(Z_i')^\Delta$ and a value $(Z_0')^\Delta \cdot s$, in which the uniform value $s \in H$ blinds $h^{f'(0)}$. That means that, information-theoretically, the adversary sees t evaluations $f'(id_i^*)$ of a polynomial f' that has t degrees of freedom (through the z_i). Hence, D prepares a random ciphertext C distributed exactly as $C_{\mathsf{pv}}^{\mathcal{R}}$ from (5.4). Thus,

$$\begin{aligned} &\Pr\left[D(1^k, n, g, u_1, g^y, Z) = 1 \mid Z = u_1^y h\right] \\ =\ &\Pr\left[\mathcal{B}(C_{\mathsf{pv}}^{\mathcal{R}}) = \mathsf{Dec}(id^*, usk_{id^*}, C_{\mathsf{pv}}^{\mathcal{R}})\right] \\ =\ &\mathrm{RQ}_{\mathcal{B},\mathcal{R}}^{id^*}. \end{aligned} \tag{5.10}$$

Taking (5.9) and (5.10) together shows (5.6) as desired. □

Claim 5.3.3 essentially says that the pirate box $\mathcal{B}$ decrypts even malformed, random ciphertexts just as decryption with the user key usk_{id^*} for the traitor identity id^* would. It remains to prove that this decryption really uniquely identifies the traitor id^*.

Claim 5.3.4. *For any fixed $pk, id^*, \mathcal{R}$, and any identity $id' \notin \mathcal{R} \cup \{id^*\}$, we have*

$$\mathrm{RQ}^{id'}_{\mathcal{B},\mathcal{R}} \leq 1 - \mathrm{Q}_{\mathcal{B},\mathcal{R}} + \varepsilon_{\mathsf{G}}, \tag{5.11}$$

for negligible ε_{G}.

Proof. We will prove that for any $pk, id^*, \mathcal{R}, id'$ as above, we have that

$$\Pr\left[\mathsf{Dec}(id^*, usk_{id^*}, C^{\mathcal{R}}_{\mathsf{pv}}) = \mathsf{Dec}(id', usk_{id'}, C^{\mathcal{R}}_{\mathsf{pv}})\right] \qquad \text{is negligible,} \tag{5.12}$$

where the probability is over a random $C^{\mathcal{R}}_{\mathsf{pv}}$ as in (5.4). From (5.12), we can deduce (5.11) by a union bound on the events that

$$\mathcal{B}(C^{\mathcal{R}}_{\mathsf{pv}}) = \mathsf{Dec}(id^*, usk_{id^*}, C^{\mathcal{R}}_{\mathsf{pv}})$$

and

$$\mathsf{Dec}(id^*, usk_{id^*}, C^{\mathcal{R}}_{\mathsf{pv}}) \neq \mathsf{Dec}(id', usk_{id'}, C^{\mathcal{R}}_{\mathsf{pv}}).$$

To show (5.12), recall that (honest) decryption under secret key usk_{id^*} computes K through a Lagrange interpolation in the exponent and postprocessing. In particular, observe that upon input a random ciphertext

$$C^{\mathcal{R}}_{\mathsf{pv}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{z_{id}})_{id \in \mathcal{R}}, u_1^{f(0)} h^{z_0}),$$

decryption will output $\mathsf{G}^{\mathsf{eddh}}_{G,H}(h^{z_0 - f^*(0)})$, for the unique degree-$\leq t$ polynomial f^* with $f^*(id) = z_{id}$, for $id \in \mathcal{R}$ and $f^*(id^*) = 0$. (We have $f^*(id^*) = 0$ since decryption uses $u_1^{usk_{id^*}} = u_1^{usk_{id^*}} \cdot h^0$ for interpolation.) Analogously, decryption under secret key $usk_{id'}$ yields $\mathsf{G}^{\mathsf{eddh}}_{G,H}(h^{z_0 - f'(0)})$, for the unique polynomial f' with $f'(id) = z_{id}$, for $id \in \mathcal{R}$ and $f'(id') = 0$. Since $id^* \neq id'$, we have $f^*(0) \neq f'(0)$, except with probability $1/n$. Thus, by the pseudorandomness of $\mathsf{G}^{\mathsf{eddh}}_{G,H}$, it follows that

$$\mathsf{G}^{\mathsf{eddh}}_{G,H}(h^{z_0 - f^*(0)}) \neq \mathsf{G}^{\mathsf{eddh}}_{G,H}(h^{z_0 - f'(0)}),$$

except with negligible probability ε_{G}. This shows the claim. □

Claim 5.3.4 upper bounds the probability that the decryption under the "wrong" identity yields the "right" result by accident. In particular, if we take $\mathrm{Q}_{\mathcal{B},\mathcal{R}} \geq 2/3$ in (5.11) and (5.6), we get

$$\mathrm{RQ}^{id^*}_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^{id'}_{\mathcal{B},\mathcal{R}} \;\geq\; 1/3 - \mathsf{Adv}^{\mathsf{eddh}}_{G,H,D}(k) - \varepsilon_{\mathsf{G}} \qquad \text{for all } id' \notin \mathcal{R} \cup \{id^*\}.$$

For the approximations $\widetilde{\mathrm{RQ}}^{id}_{\mathcal{B},\mathcal{R}}$ of $\mathrm{RQ}^{id}_{\mathcal{B},\mathcal{R}}$ computed by Trace, this implies

$$\widetilde{\mathrm{RQ}}^{id^*}_{\mathcal{B},\mathcal{R}} - \widetilde{\mathrm{RQ}}^{id'}_{\mathcal{B},\mathcal{R}} \;\geq\; 1/9 - \mathsf{Adv}^{\mathsf{eddh}}_{G,H,D}(k) - \varepsilon_{\mathsf{G}} \qquad \text{for all } id' \notin \mathcal{R} \cup \{id^*\}, \tag{5.13}$$

with overwhelming probability over the approximations. In particular, (5.13) implies that id^* maximizes $\widetilde{\mathrm{RQ}}^{id}_{\mathcal{B},\mathcal{R}}$ for sufficiently large k. Hence, if $\mathrm{Q}_{\mathcal{B},\mathcal{R}} \geq 2/3$, and $\mathsf{Adv}^{\mathsf{eddh}}_{G,H,D}(k) \leq 1/9 - \varepsilon_{\mathsf{G}}$, and all the approximations are accurate in the sense of (5.5), then Trace outputs id^*. □

5.3.2 $((t+1)/2, \varepsilon)$-Sid-traceability of RKEM

Why our Tracing Strategy for $T = 1$ does not Work. First, observe that our concrete tracing strategy from the proof of Theorem 5.3.2 fails if A requests multiple user keys. For instance, A could use multiple user keys to distinguish valid from random ciphertexts (which would break Claim 5.3.3). Concretely, A could request two keys usk_{id_1} and usk_{id_2} and let $\mathcal{B}$ first check if a given ciphertext decrypts to the same value under both usk_{id_1} and usk_{id_2}. If the decryptions do not match, then $\mathcal{B}$ immediately fails. (Recall that our proof uses the fact that random ciphertexts decrypt differently under different keys.) Such a box $\mathcal{B}$ would be useless to our tracing algorithm Trace, since Trace feeds $\mathcal{B}$ only random ciphertexts. (See [KY01b] for more details.)

How to adapt our strategy. A natural way to adapt our strategy — this essentially follows the "black-box confirmation argument" from [BF99] — would seem as follows. Given a set $I \subseteq \mathcal{ID}$ of identities, we can construct "pseudo-valid ciphertexts" of the form

$$C^{\mathcal{R},I}_{\mathsf{pv}} = (\mathcal{R}, u_1, (u_1^{f(id)} h^{f'(id)})_{id \in \mathcal{R}}, u_1^{f(0)} h^{f'(0)}), \tag{5.14}$$

for $f'(x) \in \mathbb{Z}_q[x]$ uniform of degree $\leq t$, but subject to $f'(id) = 0$ for $id \in I$. We will also define the random quality $\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}}$ of a box $\mathcal{B}$ relative to a given revoked set $\mathcal{R}$, and an identity set $I \subseteq \mathcal{ID}$:

$$\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}} := \Pr\left[\mathcal{B}(C^{\mathcal{R},I}_{\mathsf{pv}}) = \mathsf{Dec}(id, usk_{id}, C^{\mathcal{R}}_{\mathsf{pv}})\right], \tag{5.15}$$

for some $id \in I$. Intuitively, ciphertexts $C^{\mathcal{R},I}_{\mathsf{pv}}$ look consistent from the point of a pirate box that only knows user keys for identities in I. Hence, our tracing strategy for a larger number T of traitors will be as follows. We iterate over all $\binom{N}{T}$ identity subsets $I \subseteq \mathcal{ID}$ of size T, and approximate $\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}}$. If the approximation indicates that $\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}} \geq \varepsilon$, then we have a candidate for the set $\mathcal{C}$ of traitors. Unfortunately, there may be many candidates, and not all of them contain only traitors. To filter out one identity that surely is a traitor, we remove identities from I, one at a time. If the quality $\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}}$ drops, we must have removed a traitor. (If the removed identity was no traitor, then $\mathcal{B}$ would not have noticed.) Again, this tracing strategy is similar to that of [BF99, NNL01, KY01a, TT01, DFKY05, BSW06]. More formally:

Theorem 5.3.5 (($(t+1)/2, \varepsilon$)-sid-traceability of RKEM)**.** *Assuming EDDH,* RKEM *is* (T, ε)*-sid-traceable for every* $T \leq (t+1)/2$ *for which* $\binom{N}{T}$ *is polynomial, and every significant* ε*. The corresponding tracing algorithm* Trace *runs for* $\mathbf{O}(k\binom{N}{T}/\varepsilon^2)$ *steps, where* N *denotes the number of identities in the system. Concretely, for every* T*-valid adversary* A*, there are adversaries* D, E, F*, such that*

$$\left|\mathsf{Adv}^{\mathsf{trace}}_{\mathsf{RKEM},A}(k)\right| \;\leq\; \mathbf{O}(2^{-k}),$$

for all k *that satisfy*

$$\left|\mathsf{Adv}^{\mathsf{eddh}}_{G,H,D}(k)\right| + \left(\sum_{i=2}^{T}\binom{N}{i}\right)\cdot\left|\mathsf{Adv}^{\mathsf{eddh}}_{G,H,E}(k)\right| + (N-T)\cdot\left|\mathsf{Adv}^{\mathsf{eddh}}_{G,F}(k)\right| \;\leq\; \frac{\varepsilon}{3T}.$$

Proof. Fix T and $\varepsilon = \varepsilon(k)$ as above.

The tracing algorithm. First, $\mathsf{Trace}^{\mathcal{B}(\cdot)}(msk, \mathcal{R})$ iterates over all identity sets $I \subseteq \mathcal{ID}$ of size T and approximates the random quality $\mathrm{RQ}^I_{\mathcal{B},\mathcal{R}}$ (as defined in (5.15)). Again, a standard argument shows that with $\mathbf{O}(k/\varepsilon^2)$ $\mathcal{B}$-queries *for each* I, we obtain approximations $\widetilde{\mathrm{RQ}}^I_{\mathcal{B},\mathcal{R}}$ such that

$$\text{for all } I\text{:}\quad \left|\widetilde{\mathrm{RQ}}^I_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^I_{\mathcal{B},\mathcal{R}}\right| < \frac{\varepsilon}{3T}, \tag{5.16}$$

except with probability $\mathbf{O}(2^{-k})$. If no I with $\widetilde{\mathrm{RQ}}^{I}_{\mathcal{B},\mathcal{R}} > \varepsilon - \varepsilon/(3T)$ is found, Trace halts with output "`fail`". Otherwise, let $I = \{id_1, \ldots, id_T\}$ be such an I, and write $I_i := \{id_i, \ldots, id_T\}$. Now Trace approximates the values $\mathrm{RQ}^{I_i}_{\mathcal{B},\mathcal{R}}$ (for $1 \le i \le T$) as in (5.16). Finally, Trace outputs id_i for the smallest i that meets

$$\left|\widetilde{\mathrm{RQ}}^{I_i}_{\mathcal{B},\mathcal{R}} - \widetilde{\mathrm{RQ}}^{I_{i+1}}_{\mathcal{B},\mathcal{R}}\right| > \frac{\varepsilon}{T} \tag{5.17}$$

(or id_T if (5.17) holds for no $i < T$).

Why tracing works. To analyze Trace, consider an adversary A in the (T, ε)-sid-traceability experiment. We assume without loss of generality that A always requests a set $\mathcal{C}$ of exactly T user keys, and finally outputs a set $\mathcal{R} = \{id_1^*, \ldots, id_t^*\}$, along with a pirate box $\mathcal{B}$.

Our first claim essentially states that tracing does not output "`fail`" (except with small probability):

Claim 5.3.6. *There is a EDDH distinguisher D whose runtime is essentially that of the sid-traceability experiment with A, such that*

$$\mathrm{Q}_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^{\mathcal{C}}_{\mathcal{B},\mathcal{R}} = \mathsf{Adv}^{\mathsf{eddh}}_{G,H,D}(k). \tag{5.18}$$

Proof. We proceed as in the proof of Claim 5.3.3. First, D obtains $\mathcal{C}$ from A. Then D prepares a public key pk and user keys $(usk_{id})_{id \in \mathcal{C}}$ for A, and a ciphertext C for B, such that

- if $b = 0$, then C is an honest encryption, and
- if $b \neq 0$, then C is distributed as $C^{\mathcal{R},\mathcal{C}}_{\mathsf{pv}}$.

(Note that Claim 5.3.3 can be seen as the special case $\mathcal{C} = \{id^*\}$.) Finally, D outputs 1 if and only if $\mathcal{B}(C) = \mathsf{Dec}(id, usk_{id}, C)$ for some $id \in \mathcal{C}$. The analysis of D is analogous to that from Claim 5.3.3. □

Next, we show that a pirate box $\mathcal{B}$ does not notice if we remove an identity $id' \notin \mathcal{C}$ from the set I in $C^{\mathcal{R},I}_{\mathsf{pv}}$:

Claim 5.3.7. *There is a EDDH distinguisher E whose runtime is essentially that of the sid-traceability experiment with A, such that*

$$\mathrm{RQ}^{I}_{\mathcal{B},\mathcal{R}} - \mathrm{RQ}^{I\setminus\{id'\}}_{\mathcal{B},\mathcal{R}} = \left(\sum_{i=2}^{T} \binom{N}{i}\right) \cdot \mathsf{Adv}^{\mathsf{eddh}}_{G,H,E}(k), \tag{5.19}$$

for all $I \subseteq \mathcal{ID}$ *with* $2 \leq |I| \leq T$*, and every* $id' \in I \setminus \mathcal{C}$*.*

Proof. E runs A to obtain $\mathcal{C}$, and then guesses I and id' as above uniformly. Then, E prepares a public key pk and a ciphertext C for A, such that

- D knows the user keys usk_{id} for all $id \in \mathcal{C} \cup I \setminus \{id'\}$,
- if $b = 0$, then C is distributed as $C_{\mathsf{pv}}^{\mathcal{R},I}$, and
- if $b \neq 0$, then C is distributed as $C_{\mathsf{pv}}^{\mathcal{R},I\setminus\{id\}}$.

This can be done analogously to the proof of Claim 5.3.3. We stress, however, that at this point, we use that $T \leq (t+1)/2$ to fix the implicitly defined polynomial f at all $id \in \mathcal{C} \cup I$. Finally, D outputs 1 iff $\mathcal{B}(C) = \mathsf{Dec}(id, usk_{id}, C)$ for some $id \in I \setminus \{id'\}$. The analysis of D is again analogous to that from Claim 5.3.3, and (5.19) follows through an averaging argument. □

Finally, we show that if tracing ends up with a singleton set $I = \{id\}$ (such that the random quality $\mathrm{RQ}_{\mathcal{B},\mathcal{R}}^{I}$ still is high), then we must have $id \in \mathcal{C}$.

Claim 5.3.8. *There is a EDDH adversary* D *whose runtime is essentially that of the sid-traceability experiment with* A*, such that*

$$\mathrm{RQ}_{\mathcal{B},\mathcal{R}}^{\{id'\}} = (N - T) \cdot \mathsf{Adv}_{G,H,D}^{\mathsf{eddh}}(k), \tag{5.20}$$

for all $id' \in \mathcal{ID} \setminus (\mathcal{C} \cup \mathcal{R})$*.*

Proof. D obtains $\mathcal{C}$ from A, and then guesses $id' \in \mathcal{ID}\setminus\mathcal{C}$ uniformly. Then D interprets its EDDH challenge as $g, g^{f(0)}, u_1, u_1^{f(0)}h^b$, and forms a public key pk for A (with otherwise uniform and known f) as in the proof of Claim 5.3.3. Now observe that the distributions $C_{\mathsf{pv}}^{\mathcal{R}}$ and $C_{\mathsf{pv}}^{\mathcal{R},\{id'\}}$ are identical as soon as $id' \notin \mathcal{R}$. (To see this, note that a uniform f' subject to $f'(id') = 0$ still has t degrees of freedom.) Hence, D can generate a ciphertext C with u_1 as above and $u_2 = u_1^{f(0)}h^b s$ for uniform $s \in H$ and uniformly and independently distributed τ_i. Regular decryption would decrypt C to $\mathsf{G}_{G,H}^{\mathsf{eddh}}(h^b s)$ under $usk_{id'}$. So whenever $\mathcal{B}(C)$ outputs $K = \mathsf{G}_{G,H}^{\mathsf{eddh}}(h^b s)$, D can solve its own EDDH challenge (by comparing K to $\mathsf{G}_{G,H}^{\mathsf{eddh}}(s)$), and through an averaging argument, we obtain (5.20). □

Finishing Up. We can now put the pieces together and analyze the tracing algorithm Trace. Let us assume that all approximations are suitably close

in the sense of (5.16). Then, by Claim 5.3.6, and the assumption about $\mathcal{B}$, Trace will not output "`fail`" (except with negligible probability). Besides, every time Trace finishes because (5.17) holds for an i, then Claim 5.3.7 (in contrapositive form) says that $id_i \in \mathcal{C}$ really must be a traitor. Finally, if no $i < T$ meets (5.17), then $\mathrm{RQ}_{\mathcal{B},\mathcal{R}}^{\{id_T\}}$ must be significant. Claim 5.3.8 implies that then, $id_T \in \mathcal{C}$ is a traitor. □

5.3.3 Potential Generalizations of Our Tracing Result

There are several dimensions in which one might want to improve our tracing result. We will comment on how our result can be generalized (and when a generalization seems problematic).

Full (Instead of Sid-)traceability. In case of a polynomial number of identities (which is necessary for efficient tracing anyway), Lemma 5.3.1 immediately yields:

Corollary 5.3.9 (($(t+1)/2, \varepsilon$)-traceability of RKEM)**.** *Assuming EDDH,* RKEM *is (T, ε)-traceable for every $T \leq (t+1)/2$ for which $\binom{N}{T}$ is polynomial, and every significant ε.*

Generalization to Wee's Factoring-based RKEM. [Wee11] also constructs an RKEM $\mathsf{RKEM}_{\mathsf{Fact}}$ whose t-RKEM-IND-CPA-security is based on the factoring assumption. (For convenience, we have reproduced $\mathsf{RKEM}_{\mathsf{Fact}}$ in Corollary 5.3.3.) Conceptually, RKEM and $\mathsf{RKEM}_{\mathsf{Fact}}$ are very similar. $\mathsf{RKEM}_{\mathsf{Fact}}$ works over a group $\mathbb{QR}_N^+ \subseteq \mathbb{Z}_N^*$ of size $\varphi(N)/4$ for a Blum integer N. In particular, ciphertexts are of the form

$$C = (\mathcal{R}, u, (u^{f(id)})_{id \in \mathcal{R}}),$$

for some degree-$\leq t$ polynomial $f(x) = a_0 + a_1 x + \cdots + a_t x^t \in \mathbb{Z}_{\varphi(N)/4}[X]$ implicitly given in the public key. With $\mathsf{RKEM}_{\mathsf{Fact}}$, however, we always have

$$f(0) = a_0 = 2^{-(t+1)k} \bmod \varphi(N)/4.$$

Moreover, decryption of an honestly generated ciphertext yields $\mathrm{BBS}_N(s)$ for the BBS pseudorandom generator [BBS82] and $s = u^{-2^k}$. These modifications (compared to RKEM) enable a reduction to the factoring assumption; however, they also have a number of other effects.

Specifically, given a potential raw key s, we can always check if s is the correct decryption of a (consistent) ciphertext by checking if $s^{2^k} = u$ holds. This also gives a way to distinguish completely random ciphertexts $C^{\mathcal{R}}_{\mathsf{pv}}$ from honestly generated ciphertexts. (Pseudo-valid ciphertexts $C^{\mathcal{R}}_{\mathsf{pv}}$ yield uniform values s upon decryption, which can be recognized.) This leads to problems during the proof of Claim 5.3.3. Hence, we do not even know if Wee's factoring-based scheme $\mathsf{RKEM}_{\mathsf{Fact}}$ is $(1, 2/3)$-sid-traceable.

Now, we restate Wee's construction based on the hardness of factoring. (Again, this construction is similar to the EDDH-based construction RKEM.)

Wee's Factoring-based RKEM [Wee11]. The factoring-based $\mathsf{RKEM}_{\mathsf{Fact}}$ of Wee consists of four PPT algorithms $(\mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

Key generation. $\mathsf{Gen}(k, t)$ chooses a Blum integer $N = PQ$, along with a uniform generator g of the group $\mathbb{QR}^+_N$ of signed quadratic residues.[2] Gen then chooses uniform exponents $a_i \in \mathbb{Z}_{\varphi(N)/4}$ (for $i \in [t]$) and sets

$$f(x) := 2^{-(t+1)k} + a_1 x + \ldots a_t x^t \bmod \varphi(N)/4.$$

The output is

$$pk := (N, g, (g^{a_i 2^{(t+1)k}})_{i=1}^t) \quad \text{and} \quad msk := (P, Q, (a_i)_{i=1}^t).$$

Key extraction. $\mathsf{Ext}(msk, id)$, for $id \in [\sqrt{N}/4]$, returns

$$usk_{id} := f(id) \bmod \varphi(N)/4.$$

Encapsulation. $\mathsf{Enc}(pk, \mathcal{R})$ chooses an exponent[3] $r \leftarrow \mathbb{Z}_{\lceil N/4 \rceil}$, and com-

[2] If we write $\mathbb{Z}_N = \{-(N-1)/2, \ldots, (N-1)/2\}$, and denote with $\mathbb{J}_N \subseteq \mathbb{Z}_N$ all elements with Jacobi symbol 1, then $\mathbb{QR}^+_N = \{|x| : x \in \mathbb{J}_N\}$. When letting $x \cdot y := |xy|$ for $x, y \in \mathbb{QR}^+_N$, then $\mathbb{QR}^+_N$ is isomorphic to the group $\mathbb{QR}_N$ of quadratic residues modulo N. In particular, $|\mathbb{QR}^+_N| = \varphi(N)/4$. However, unlike $\mathbb{QR}_N$, $\mathbb{QR}^+_N$ is efficiently recognizable, which can be advantageous in some cases. See [HK09] for details and further references.

[3] While Enc cannot choose a uniform exponent via $r \leftarrow \mathbb{Z}_{\varphi(N)/4}$, choosing $r \leftarrow \mathbb{Z}_{\lceil N/4 \rceil}$ is statistically close.

putes

$$u := g^{r2^{(t+1)k}}, \quad s := g^{r2^{tk}} \quad \big(= u^{2^{-k}} = u^{f(0)\cdot 2^{tk}} \big),$$
$$h_{id} := \left(g \cdot \prod_{i=1}^{t} \left(g^{a_i 2^{(t+1)k}} \right)^{id^i} \right)^r \quad \big(= u^{f(id)} \big).$$

(for $id \in \mathcal{R}$). Ciphertext is

$$C := (\mathcal{R}, u, (h_{id})_{id\in\mathcal{R}}),$$

and key is $K := \mathrm{BBS}_N(s)$, where $\mathrm{BBS}_N(s)$ is the BBS pseudorandom generator [BBS82] applied to s and modulo N.

Decapsulation. $\mathsf{Dec}(id, usk_{id}, C)$, with $usk_{id} = f(id)$ and C as above, sets

$$h_{id} := u^{f(id)\cdot 2^{(t+1)k}}$$

and then retrieves

$$s := u^{f(0)\cdot 2^{tk}}$$

from the h_{id} through Lagrange interpolation in the exponent. Note that this has to be done via a "gcd in the exponent" argument (see [Sha81]), since decryption cannot compute the fractional Lagrange coefficients directly. (This also explains the slightly tedious additional 2^{tk}-factor in the exponent; we refer to [Wee11] for details.)

Chapter 6

Conclusion and Open Problems

In this work, we consider three aspects of cryptographic building blocks and transformations. We give a brief conclusion.

Confined Guessing. In the digital signature context, we present the concept of confined guessing. This technique is developed two-staged. First, we define the cryptographic building block tag-based signatures (where a signature additionally includes a tag) together with a milder security notion in comparison to the standard security notion for signatures. (The idea is that this milder security notion can easier be achieved by tag-based signature instantiations.) Secondly, we give a cryptographic transformation from several only mildly secure tag-based signatures to standard secure signatures with compact parameters. Within the transformation, we make use of the confined guessing concept. Concretely, we partition the tag space in the transformation such that there exist a large enough (but not too large) set. From this confined set, we are able to guess tags with significant probability which is helpful in the transformation. A downside of the transformation might be that it possesses a large loss. Hence, an open problem would be to reduce this loss.

(Almost) Tight IBE security. We consider and present a tight transformation in the IBE context. (IBE can be seen as an extended form of encryption, where any public identifier and some public parameter suffice to encrypt a message.) Common IBE security notions usually involve only one instance and one chiphertext. Real-world scenarios, however, often consider multiple

instances with multiple ciphertexts. It is known that one can trivially transform standard secure IBE systems in the one-instance, one-ciphertext setting to the multi-instance, multi-ciphertext case. Unfortunately, within this cryptographic transformation, the security guarantees degrade in the number of instances and ciphertexts. In [CW13], Chen and Wee propose an (almost) tightly secure IBE system in the one-instance, one-ciphertext setting. We extend their underlying building block and propose an (almost) tightly secure IBE in the multi-instance, multi-ciphertext setting. Concretely, we give a cryptographic transformation from the security of our IBE to the security of the underlying building block in the multi-instance, multi-ciphertext setting. Concerning open problems, we mention that in the reduction, we need an additional assumption to prove a very strong form of IBE security in the multi-instance, multi-ciphertext setting. An open problem would be: is it possible to extend the underlying building block further to prove an IBE strongly secure in multi-instance, multi-ciphertext setting without relying on the additional assumption?

A Generic View on Trace-and-Revoke Systems. Trace-and-revoke systems yield a strong cryptographic tool to fight piracy in the area of content protection. We give a new generic view of trace-and-revoke instantiations under a simple generic assumption. Concretely, we first extend a generic work by Wee [Wee11] by giving a slightly different generic view. Essentially, we connect a generic assumption due to Hemenway and Ostrovsky [HO12] with the work from Wee. In particular, this yields a new generic view of revocation systems. Secondly, we show that the emerged revocation instantiations are traceable (i.e., they allow a form of catching traitors). Hence, we derive trace-and-revoke instantiations. In particular, this yields a new generic view on trace-and-revoke instantiations under a simple generic assumption. Put together, we give a new generic view of trace-and-revoke systems that generalizes known and new trace-and-revoke instantiations. An open problem would be to generalize our result further; e.g., a trace-and-revoke instantiation under the factoring assumption is currently, to the best of our knowledge, unknown. (In particular, Wee provides factoring-based revocation instantiations, but it is unknown if those instantiations are traceable.)

References

[AB09] Shweta Agrawal and Xavier Boyen. Identity-based encryption from lattices in the standard model. Manuscript, July 2009. Available at `http://www.cs.stanford.edu/~xb/ab09/`.

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, May 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, August 2010.

[ABV+12] Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee. Functional encryption for threshold functions (or fuzzy ibe) from lattices. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 280–297. Springer, May 2012.

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, May 2004.

[BB04b] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459. Springer, August 2004.

[BB04c] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, May 2004.

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, May 2000.

[BBS82] Lenore Blum, Manuel Blum, and Mike Shub. Comparison of two pseudo-random number generators. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 61–78. Plenum Press, New York, USA, 1982.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th FOCS*, pages 514–523. IEEE Computer Society Press, October

1996.

[BDJR97] Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403. IEEE Computer Society Press, October 1997.

[BF99] Dan Boneh and Matthew K. Franklin. An efficient public key traitor tracing scheme. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 338–353. Springer, August 1999.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, August 2001.

[BF03] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003. extended abstract in Crypto'01.

[BG90] Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 194–211. Springer, August 1990.

[BGR95] Mihir Bellare, Roch Guérin, and Phillip Rogaway. XOR MACs: New methods for message authentication using finite pseudorandom functions. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 15–28. Springer, August 1995.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, August 2005.

[BH08] Dan Boneh and Michael Hamburg. Generalized identity based and broadcast encryption schemes. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 455–470. Springer, December 2008.

[BH12] Itay Berman and Iftach Haitner. From non-adaptive to adaptive pseudorandom functions. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 357–368. Springer, March 2012.

[BH15] Itay Berman and Iftach Haitner. From non-adaptive to adaptive pseudorandom functions. *Journal of Cryptology*, 28(2):297–311, April 2015.

[BHJ+13] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, Jae Hong Seo, and Christoph Striecks. Practical signatures from standard assumptions. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 461–485. Springer, May 2013.

[BHJ+15] Florian Böhl, Dennis Hofheinz, Tibor Jager, Jessica Koch, and Christoph Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, 28(1):176–208, January 2015.

[BKP14] Olivier Blazy, Eike Kiltz, and Jiaxin Pan. (hierarchical) identity-based encryption from affine message authentication. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 408–425. Springer, August 2014.

[BKR94] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of cipher block chaining. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 341–358. Springer, August 1994.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514–532. Springer, December 2001.

[BM88] Mihir Bellare and Silvio Micali. How to sign given any trapdoor function (extended abstract). In *20th ACM STOC*, pages 32–42. ACM Press, May 1988.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, May 1996.

[BR09] Mihir Bellare and Thomas Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 407–424. Springer, April 2009.

[BSW06] Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 573–592. Springer, May / June 2006.

[BW06] Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06*, pages 211–220. ACM Press, October / November 2006.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300. Springer, December 2013.

[BWZ14] Dan Boneh, Brent Waters, and Mark Zhandry. Low overhead broadcast encryption from multilinear maps. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 206–223. Springer, August 2014.

[CD95] Ronald Cramer and Ivan Damgård. Escure signature schemes based on interactive protocols. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 297–310. Springer, August 1995.

[CD96] Ronald Cramer and Ivan Damgård. New generation of secure and practical RSA-based signatures. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 173–185. Springer, August 1996.

[CFN94] Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 257–270. Springer, August 1994.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271. Springer, May 2003.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, May 2004.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, May 2010.

[CLL+13] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter IBE and signatures via asymmetric pairings. In Michel Abdalla and Tanja Lange, editors, *PAIRING 2012*, volume 7708 of *LNCS*, pages 122–140. Springer, May 2013.

[CLL+14] Jie Chen, HoonWei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter identity-based encryption via asymmetric pairings. *Designs, Codes and Cryptography*, 73(3):911–947, 2014.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363, London, UK, UK, 2001. Springer-Verlag.

[Cor00] Jean-Sébastien Coron. On the exact security of full domain hash. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 229–235. Springer, August 2000.

[CPP05] Hervé Chabanne, Duong Hieu Phan, and David Pointcheval. Public

traceability in traitor tracing schemes. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 542–558. Springer, May 2005.

[CS99] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. In *ACM CCS 99*, pages 46–51. ACM Press, November 1999.

[CW13] Jie Chen and Hoeteck Wee. Fully, (almost) tightly secure IBE and dual system groups. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 435–460. Springer, August 2013.

[CW14] Jie Chen and Hoeteck Wee. Dual system groups and its applications — compact HIBE and more. Cryptology ePrint Archive, Report 2014/265, 2014. `http://eprint.iacr.org/2014/265`.

[Del07] Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 200–215. Springer, December 2007.

[DF02] Yevgeniy Dodis and Nelly Fazio. Public key broadcast encryption for stateless receivers. In Joan Feigenbaum, editor, *Digital Rights Management Workshop*, volume 2696 of *Lecture Notes in Computer Science*, pages 61–80. Springer, 2002.

[DF03] Yevgeniy Dodis and Nelly Fazio. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 100–115. Springer, January 2003.

[DFKY05] Yevgeniy Dodis, Nelly Fazio, Aggelos Kiayias, and Moti Yung. Scalable public-key tracing and revoking. *Distributed Computing*, 17(4):323–347, 2005.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DM14] Léo Ducas and Daniele Micciancio. Improved short lattice signatures in the standard model. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 335–352. Springer, August 2014.

[DN94] Cynthia Dwork and Moni Naor. An efficient existentially unforgeable signature scheme and its applications. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 234–246. Springer, August 1994.

[DQ87] Yvo Desmedt and Jean-Jacques Quisquater. Public-key systems based

on the difficulty of tampering (is there a difference between DES and RSA?). In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 111–117. Springer, August 1987.

[DR06] T. Dierks and E. Rescorla. The transport layer security (tls) protocol. In *IETF RFC 4346*, 2006.

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, August 1984.

[FN94] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer, August 1994.

[FP12] Nelly Fazio and Irippuge Milinda Perera. Outsider-anonymous broadcast encryption with sublinear ciphertexts. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 225–242. Springer, May 2012.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464. Springer, May / June 2006.

[GH09] Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 437–456. Springer, March 2009.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

[GMY83] Shafi Goldwasser, Silvio Micali, and Andrew Chi-Chih Yao. Strong signature schemes. In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 431–439. ACM, 1983.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography.

In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 548–566. Springer, December 2002.

[GW09] Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 171–188. Springer, April 2009.

[HJ12] Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 590–607. Springer, August 2012.

[HJK11] Dennis Hofheinz, Tibor Jager, and Eike Kiltz. Short signatures from weaker assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 647–666. Springer, December 2011.

[HJW00] Detlef Hühnlein, Jr Jacobson, MichaelJ., and Damian Weber. Towards practical non-interactive public key cryptosystems using non-maximal imaginary quadratic orders (extended abstract). In DouglasR. Stinson and Stafford Tavares, editors, *Selected Areas in Cryptography*, volume 2012 of *Lecture Notes in Computer Science*, pages 275–287. Springer Berlin Heidelberg, 2000.

[HK04] Swee-Huay Heng and Kaoru Kurosawa. k-resilient identity-based encryption in the standard model. In Tatsuaki Okamoto, editor, *CT-RSA 2004*, volume 2964 of *LNCS*, pages 67–80. Springer, February 2004.

[HK08] Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 21–38. Springer, August 2008.

[HK09] Dennis Hofheinz and Eike Kiltz. The group of signed quadratic residues and applications. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 637–653. Springer, August 2009.

[HKS15] Dennis Hofheinz, Jessica Koch, and Christoph Striecks. Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 799–822. Springer, March / April 2015.

[HL02] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 466–481. Springer, April / May 2002.

[HO12] Brett Hemenway and Rafail Ostrovsky. Extended-DDH and lossy trapdoor functions. In Marc Fischlin, Johannes Buchmann, and Mark

Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 627–643. Springer, May 2012.

[HS02] Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 47–60. Springer, August 2002.

[HS14] Dennis Hofheinz and Christoph Striecks. A generic view on trace-and-revoke broadcast encryption schemes. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 48–63. Springer, February 2014.

[HW09a] Susan Hohenberger and Brent Waters. Realizing hash-and-sign signatures under standard assumptions. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 333–350. Springer, April 2009.

[HW09b] Susan Hohenberger and Brent Waters. Short and stateless signatures from the RSA assumption. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 654–670. Springer, August 2009.

[JR13] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, December 2013.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS 2000*. The Internet Society, February 2000.

[KY01a] Aggelos Kiayias and Moti Yung. On crafty pirates and foxy tracers. In *Digital Rights Management Workshop*, pages 22–39, 2001.

[KY01b] Aggelos Kiayias and Moti Yung. Self protecting pirates and black-box traitor tracing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 63–79. Springer, August 2001.

[KY02] Aggelos Kiayias and Moti Yung. Traitor tracing with constant transmission rate. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 450–465. Springer, April / May 2002.

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979.

[Lew12] Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 318–335. Springer, April 2012.

[LJYP14] Benoît Libert, Marc Joye, Moti Yung, and Thomas Peters. Concise multi-challenge CCA-secure encryption and signatures with al-

most tight security. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 1–21. Springer, December 2014.

[LL93] ChaeHoon Lim and PilJoong Lee. Modified maurer-yacobi's scheme and its applications. In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology — AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 308–323. Springer Berlin Heidelberg, 1993.

[LPQ12] Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 206–224. Springer, May 2012.

[LSW10] Allison B. Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *2010 IEEE Symposium on Security and Privacy*, pages 273–285. IEEE Computer Society Press, May 2010.

[LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, February 2010.

[Mer88] Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 369–378. Springer, August 1988.

[Mer90] Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 218–238. Springer, August 1990.

[MH78] Ralph C. Merkle and Martin E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, 24(5):525–530, 1978.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, April 2012.

[MY91] Ueli M. Maurer and Yacov Yacobi. Non-interactive public-key cryptography. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 498–507. Springer, April 1991.

[MY93] Ueli M. Maurer and Yacov Yacobi. A remark on a non-interactive public-key distribution system (rump session). In Rainer A. Rueppel, editor, *EUROCRYPT'92*, volume 658 of *LNCS*, pages 458–460.

Springer, May 1993.

[MY96] UeliM. Maurer and Yacov Yacobi. A non-interactive public-key distribution system. *Designs, Codes and Cryptography*, 9(3):305–316, 1996.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, August 2001.

[NP98] Moni Naor and Benny Pinkas. Threshold traitor tracing. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 502–517. Springer, August 1998.

[NP01] Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In Yair Frankel, editor, *FC 2000*, volume 1962 of *LNCS*, pages 1–20. Springer, February 2001.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

[Oka93] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, August 1993.

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 387–398. Springer, May 1996.

[Rab79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Cambridge, MA, USA, 1979.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, August 1992.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[Sha81] Adi Shamir. The generation of cryptographically strong pseudo-random sequences. In Allen Gersho, editor, *CRYPTO'81*, volume ECE

Report 82-04, page 1. U.C. Santa Barbara, Dept. of Elec. and Computer Eng., 1981.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, August 1984.

[Sho00] Victor Shoup. Practical threshold signatures. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 207–220. Springer, May 2000.

[Tan88] Hatsukazu Tanaka. A realization scheme for the identity-based cryptosystem. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 340–349. Springer, August 1988.

[TI89] S. Tsujii and T. Itoh. An id-based cryptosystem based on the discrete logarithm problem. *IEEE J.Sel. A. Commun.*, 7(4):467–473, May 1989.

[TT01] Wen-Guey Tzeng and Zhi-Jia Tzeng. A public-key traitor tracing scheme with revocation using dynamic shares. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 207–224. Springer, February 2001.

[Wat05] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, May 2005.

[Wat09] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, August 2009.

[Wee10] Hoeteck Wee. Efficient chosen-ciphertext security via extractable hash proofs. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 314–332. Springer, August 2010.

[Wee11] Hoeteck Wee. Threshold and revocation cryptosystems via extractable hash proofs. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 589–609. Springer, May 2011.

[Wee14] Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, February 2014.

Curriculum Vitae

01/14/1984	Born in Salzwedel
1994 – 2003	Abitur (Käthe-Kollwitz-Gymnasium Salzwedel)
2003 – 2004	Civilian service
10/2004 – 07/2010	Studies (TU Braunschweig, Degree: Diplom-Informatiker)
03/2008 – 09/2008	Internship (Siemens Coorporate Research, Princeton, New Jersey, USA, job: software developer)
07/2010 –	Research Associate (KIT, Institute of Theoretical Informatics, Cryptography and Security)