KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING INFORMATICA
Celestijnenlaan 200 A — B-3001 Leuven

Provably Secure Identity-Based Identification Schemes and Transitive Signatures

Promotoren :
Prof. Dr. ir. F. PIESSENS
Prof. Dr. ir. B. DE DECKER

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Gregory NEVEN

May 2004

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT COMPUTERWETENSCHAPPEN
AFDELING INFORMATICA
Celestijnenlaan 200 A — B-3001 Leuven

Provably Secure Identity-Based Identification Schemes and Transitive Signatures

Jury :
Prof. Y. Willems, voorzitter
Prof. F. Piessens, promotor
Prof. B. De Decker, promotor
Prof. M. Bellare
(University of California at San Diego)
Prof. A. Haegemans
Prof. B. Preneel
Prof. J.-J. Quisquater
(Université Catholique de Louvain-la-Neuve)

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Gregory NEVEN

U.D.C. 681.3*D46

May 2004

D/2004/7515/20
ISBN 90–5682–482–1

Abstract

Cryptography is an ancient craft, but relatively young as a true science. Techniques that offered a reasonable level of protection many centuries ago, are clearly insufficient to meet the communication needs of today's digitalized society. Until the 1980s however, cryptographic design remained a craft, rather than a science: schemes were proposed with at most an intuition for their security, the sole criterion being resistance against attacks after years of exposure to experts.

A more modern approach is that of *provable security*. This approach requires the designer of a scheme to first clearly state what is understood under the security of the scheme. Next, a mathematical proof is needed showing that the only way to break the scheme is either by attacking an insecure underlying cryptographic building block, or by realizing a mathematical breakthrough. Provable security has evolved from a toy for theoreticians to an important scheme characteristic that is taken into account in the decision of industry standards.

In this thesis, we study the provable security of selected cryptographic primitives. We first distill useful yet feasible security notions, and subsequently prove the security of existing and new schemes under these notions.

The first part focuses on *identity-based identification* and *signature schemes*. These are cryptographic primitives providing entity and message authentication, respectively, that allow the public key of a user to be simply his identity (instead of a random string that has to be securely attributed to the user). As a first step, we present a general framework of security-preserving transformations between related primitives. We then use this framework as a tool to prove (and in a single instance, break) the security of schemes from 13 different "families" that were proposed in the literature over the last two decades, but that lacked a security proof prior to our work.

In the second part of this thesis, we discuss *transitive signature schemes*. These are signature schemes that allow to sign edges of a graph such that any user (and not just the signer) can, from two signatures on adjacent edges $\{i, j\}$ and $\{j, k\}$, compute a third signature for the direct edge $\{i, k\}$. We answer an open question regarding the security of a particular scheme, and present a number of new, provably secure schemes offering efficiency advantages over existing schemes.

Acknowledgements

Looking back at the three and a half years that I spent as a Ph.D student, I realize that there is no way of denying the impact they had on both my career and on me as a person. I have learned things, been to places, met people and discovered aspects of myself that I had never dreamed of. These years have been a thrilling experience of which I wouldn't have wanted to miss a single second. My sincere gratitude goes out to everyone who made this possible.

First of all, I would like to thank my advisors Prof. Frank Piessens and Prof. Bart De Decker for giving me the chance to pursue a Ph.D in cryptography. As my Master's thesis supervisor (or should I say coach), Frank guided me on my first steps in the field, and he was a driving force in getting my first papers published. Many thanks also for all the career advice, and for the confidence that this text would finally come to be, especially in times when my own was rather low. I also would like to thank Prof. Ann Haegemans and Prof. Bart Preneel for proof-reading this text, Prof. Jean-Jacques Quisquater and Prof. Mihir Bellare for accepting to be members of the jury, and Prof. Yves Willems for chairing the jury.

This Ph.D was generously made possible by a Research Assistantship from the Fund for Scientific Research – Flanders (F.W.O.–Vlaanderen), and by two much appreciated travel grants from the same organization covering the expenses of my visits to UCSD. Thanks also to Prof. Lars Knudsen, Prof. Phillip Rogaway and Prof. Jean-Jacques Quisquater for acting as references in the application for these travel grants, and to Prof. Frank Piessens and Prof. Pierre Verbaeten for repeatedly finding funds to attend summer schools and conferences around the world.

But if there is one person that I am truly indebted to for bringing this Ph.D to a successful end, then it must be Prof. Mihir Bellare. The entire technical content of this thesis is based on research done under his guidance during two visits at UCSD. The amount of time and energy he spends with his students is simply amazing, and I am very grateful to him for fostering me as one of his own students. Chapter 3 in this thesis is joint work with Dr. Chanathip (Meaw) Namprempre, and would never have made it to a Eurocrypt publication without

her perseverance; thanks for believing in the project, Meaw. I also want to thank Dr. Marc Fischlin for interesting discussions and pointing out that the $\mathcal{Sh}$-$\mathcal{SI}$ scheme is zero-knowledge, and Prof. Bart Preneel for his inspiring remarks on this chapter.

Then there are all those people that make the difference between *living* my life and *enjoying* it: my Friends. I consider myself extremely fortunate to be able to say that there are so many of You that I decided to not even attempt summarizing You all by name. I hope You will forgive me for not doing so; I decided that the injustice done by not including names does not compare to the risk of forgetting a single one. Not in the very least does this imply that I underestimate the value of Your Friendship; to the contrary, it is most precious to me. I especially want to thank everyone at volleyball club Sparvoc Lanaken for putting up with a freelance "strandjeannette"; friends and colleagues from K.U.Leuven and QMedit for putting up with an on-and-off colleague and friend; friends in Leuven and surroundings for keeping the Art of the Party alive; labmates, colleagues, surf buddies, volleyball team mates and other friends in California for welcoming me in a stimulating environment, helping me out with settling issues, and making my stays in San Diego unforgettable. Thank you all guys; you make life worthwhile.

Finally, I'd like to thank my parents, my sister Marilyn and my brother Christopher for all the fun, for the continuous support, for encouraging me to follow my ambitions around the world, and for following me around the world on their holidays.

To my grandfather, Dr. René Neven.
For I have yet to meet a more widely interested man.

Contents

List of Figures

List of Symbols and Abbreviations

The following is a list of symbols and abbreviations used throughout this text. The numbers after the descriptions refer to the pages where the concepts or notations are introduced.

$\{0,1\}^*$	set of all bit strings 11
1^k	bit string of k ones 11
$\|$	concatenation of bit strings 11
$\lvert\cdot\rvert$	bit string length, set cardinality or absolute value 11
$\leftarrow$	assignment 11
$\xleftarrow{R}$	randomized assignment or uniform selection 11
$\equiv$	equivalence
$\wedge$	logical AND operation
$\vee$	logical OR operation
$\left(\frac{a}{N}\right)$	Legendre/Jacobi symbol of a with respect to N 16

A

$\mathsf{A}(x : \textsc{Or})$	execution of algorithm A on input x with access to oracle Or 11
$[\mathsf{A}(x : \textsc{Or})]$	set of all possible outputs of randomized algorithm A on input x with access to oracle Or 11
`acc`	accept state 31
acc	accept probability function 51
$\mathbf{Adv}^{\mathrm{sec}}_{\mathcal{S},\mathsf{A}}(\cdot)$	advantage function of algorithm A in breaking scheme $\mathcal{S}$ under security notion sec 13

C

CA	certification authority 131
CDH	computational Diffie-Hellman problem 20
cert	certificate 25
Comp	composition algorithm of TS scheme 91
cSI	convertible standard identification scheme 43
cSS	convertible standard signature 47

D

DDH	decision Diffie-Hellman problem 20
DH	Diffie-Hellman 20
Dom	domain

E

$\hat{e}$	pairing function 22
ε	empty string 11
$\mathbf{Exp}^{\mathrm{sec}}_{\mathcal{S},\mathsf{A}}$	outcome of security experiment letting algorithm A attack scheme $\mathcal{S}$ under security notion sec 13

F

`false`	boolean false value

G

$\mathbb{G}$	(discrete logarithm) group 18
$\hat{\mathbb{G}}$	Gap Diffie-Hellman group 21
Gap-DH	Gap Diffie-Hellman problem 21
gcd	greatest common divisor

H

H(·)	hash function, possibly modelled as a random oracle ... 14
HQR_N	set of higher quadratic residues modulo N 57
HVZK	honest-verifier zero-knowledge 53

I

K

L

M

N

P

P	standard prover algorithm 31
$\overline{\mathsf{P}}$	identity-based prover algorithm 32
$\emptyset$	empty set
pk	public key 31
PKI	public-key infrastructure 5
$\Pr[\,\cdot\,]$	probability

Q

$\mathbf{Q}_{\mathsf{A}}^{\textsc{Or}}$	number of oracle queries of algorithm A to oracle Or .. 11
$\mathbf{Q}_{\mathsf{A}}$	total number of oracle queries of algorithm A 11
QR_N	set of quadratic residues modulo N 55

R

ρ_{A}	length of random tape of algorithm A 11
$\mathbf{R}$	trapdoor samplable relation 42
Ran	range
`rej`	reject state 31
res	reset probability function 51
RSA	Rivest, Shamir, Adleman algorithm 2

S

σ	signature
Sample	relation sampling algorithm 42
SI	standard identification scheme 31
Sign	standard signing algorithm 34
$\overline{\mathsf{Sign}}$	identity-based signing algorithm 35
sk	secret (a.k.a. private) key 31
SS	standard signature scheme 34
St	state information 12
$\mathrm{swap}(\cdot,\cdot)$	swap values of two variables

T

U

V

Z

Chapter 1

Introduction

1.1 Provably Secure Cryptography

Cryptography is an ancient craft, but relatively young as a true science. As early as 4000 years ago, the Egyptians occasionally obscured their inscriptions by deviating from standard hieroglyphic notation to make the message seem more important. The Spartans wrote secret messages along the axis of a stick with a strip of parchment spiralled around it; the parchment was unwound for transportation, so that the recipient could recover the message by rewinding it around a stick of the same diameter. Julius Caesar even has a cipher named after him that replaces every letter with that three positions further in the alphabet.

While such techniques may have offered a reasonable level of protection in times when a vast majority of the population was illiterate to begin with, they are clearly insufficient to meet today's communication needs. With the Internet as a global information infrastructure connecting an ever-increasing number of businesses, institutions and governments worldwide, information that used to take a considerable effort to retrieve is now readily available at the click of mouse. The flip side of this wonderful evolution, however, is a wider exposure to possibly malicious users, while at the same time putting higher prizes at stake. With legal protection appearing too slow and cumbersome to act as an effective deterrent in the fast-moving world of information technology, prevention of attacks by technical means is more important than ever.

Not surprisingly, public interest in cryptography boomed around the same time that digital information started to claim its place in society. The seventies saw the birth of the block cipher DES (Data Encryption Standard [Nat77]), but the real breakthrough came with the conceptual invention of public-key cryptography by Diffie and Hellman [DH76] in 1976. No longer did Alice have to meet Bob in person to agree on a common key that they could later use to

protect their conversation; instead, Diffie and Hellman suggested to use *pairs* of keys, consisting of a *public* and a *private* key. Alice would publish her public key, but keep her private key for herself. When Bob wants to send a message to Alice, he uses her public key to encrypt the message; the resulting *ciphertext* can only be decrypted using the corresponding private key, which only Alice knows. Obviously, the keys must be related in some way for this mechanism to work, but the scheme would be designed in such a way that it is infeasible to deduce the private key from the public key within a reasonable amount of time. Two years later, Rivest, Shamir and Adleman published the RSA algorithm [RSA78] as the first construction for such a public-key cryptosystem after its inventors, a contribution for which they were awarded the ACM Turing Award in 2002. Many alternative constructions based on various problems followed, but most of these were later found to be insecure. Among those that are still considered secure today (including the ElGamal [El 84], Rabin [Rab79] and Paillier [Pai99] cryptosystems), RSA remains the most widely used today.

But cryptography is not only about keeping eavesdroppers from listening in on a private conversation; it is concerned with information security in a broader sense of the word. The following is a list of four basic goals [MvOV96] envisaged by different cryptographic primitives:

- *Confidentiality*, the most well-known of the four, is to hide information from unauthorized readers.

- *Integrity* ensures that the information that reaches the recipient was not modified in transit. While it is generally not possible to *prevent* unauthorized tampering of data, cryptography does provide techniques to *detect* such behavior.

- *Authentication* provides a guarantee that entities are who they claim to be (*entity authentication*) and that messages originate from the sender they appear to originate from (*message authentication*). Although these two guarantees are closely related, they are essentially different: the former ensures the active presence of an entity at a particular instant in time, while the latter certifies who created a piece of information in the past and the fact that it has not been altered since.

- *Non-repudiation* commits an entity to his actions or statements, so that he cannot deny them at a later time.

Many primitives realizing these goals have been suggested over the past three decades. Summarizing all of them would take us far beyond the scope of this text, we refer to Menezes et al.'s *Handbook of Applied Cryptography* [MvOV96] instead. Yet still, these are not the final goals of cryptography. The above primitives can be further combined into complex protocols achieving higher-end goals

such as secure channels [FKK96, CK01, Nam02], digital cash [Cha83, JY96], electronic voting [Ben87, Sch99] and certified e-mail [SR98], to name a few.

Provable Security and the Random Oracle Model. The invention of public-key cryptography sparked an enormous interest in information security from the academic world. For at least another decade, however, the design of cryptographic primitives and protocols remained more of a craft, rather than a true science. Schemes were proposed with at most an intuition why they might be hard to break, if any at all, the only real security criterion being resistance against attacks after years of exposure to the scrutiny of experts in the field.

In the early eighties, Goldwasser and Micali [GM84] pioneered the approach of *provable security*, sometimes more appropriately called *reductionist security*, which was further brought to practice by Bellare and Rogaway during the nineties [Bel98]. The idea of provable security is to provide, along with a scheme, a mathematical proof showing that any attack on the scheme can be transformed into an attack on an underlying primitive or mathematical problem, thereby directly tying the security of the scheme to the security of its building blocks. Obviously, one cannot hope to build secure schemes out of insecure building blocks, but it is fairly easy (and unfortunately quite common) to design schemes that use the strongest cryptographic primitives around, yet are completely insecure. A security proof guarantees that any weakness in the scheme must be related to a corresponding weakness in one of the subcomponents, and that no additional vulnerabilities were introduced by the way the scheme glues the components together. While this doesn't completely exclude the scheme from being broken, it does minimize the chances of an adversary attacking it: he either has to force a mathematical breakthrough, or break an insecure subcomponent. The former type of attack is highly unlikely, the latter is easily protected against by replacing the insecure component with a secure alternative. (Two other possibilities of attacks are errors contained in the proof, and adversaries stepping out of the model. While the rigorousness of mathematical proof methodology serves as a safeguard against the first, it emphasizes the need for diligently written out security proofs and careful verification by authors and peer researchers. The second is much harder to protect against: a proof inevitably needs an adversarial model, and does not provide any guarantees against adversaries that go beyond this model. Good examples of such attacks are timing [Koc96] and power analysis [KJJ99] attacks, where the adversary extracts additional information by carefully measuring the time or energy required to perform cryptographic operations.)

Motivated by the lack of provably secure constructions that were efficient enough to replace heuristic schemes in use around the mid-nineties, Bellare and Rogaway [BR93a] suggested the *random oracle model* as a compromise between theory and practice. The idea is that security is proven in an imaginary model where all algorithms, including the adversary, have access to an oracle $\mathrm{H}(\cdot)$

that implements a random function. When brought to practice, the random oracle is replaced with a cryptographic hash function such as SHA-1 [Nat95] or RIPEMD-160 [DBP96], which is hoped to sufficiently mimic the unpredictable behavior of a true random oracle to preserve security in the real world.

A lot of controversy exists in the cryptographic community about the true value of security proofs using random oracles (we provide more details of this discussion in Section 2.2), and indeed, they should be treated with care. Theoretically speaking, the random oracle model reduces the value of a security proof to at most a good heuristic, and a number of (though mostly contrived) schemes separating it from the standard model have been discovered [CGH98, Nie02, GK03]. On the other hand, it has proven to be a most valuable tool in proving the security of new schemes and old practical schemes resisting both attack and proof in the standard model for years. Moreover, a scheme that can be proven to achieve a clear security goal in the random oracle model is still strongly preferable over completely ad-hoc protocol design. It is with this precaution in mind that we use random oracles in this work as well.

1.2 Summary and Main Contributions

In this thesis, we continue the line of provable security by targeting selected cryptographic primitives for which we first distill a useful and realistic security notion, and then proceed to prove the security of existing schemes that lacked a proof prior to our work, or of completely new schemes that offer advantages over existing ones. More specifically, this text is subdivided in two main parts. The first focuses on identity-based identification and signature schemes, for which we found a security-preserving transformation that we apply to prove the security of about a dozen schemes proposed over the last two decades and new schemes surfaced from these [1]. The second part of this thesis treats transitive signature schemes, where we answer an open question raised by [MR02b] and introduce new schemes offering considerable efficiency improvements over existing ones [3]. Some older work on secure distributed computations [8, 10, 9, 6, 5, 4, 2] and application integration [7] was not included here. We now proceed to discuss our contributions in more detail.

The results on identity-based identification schemes are joint work with Mihir Bellare and Chanathip Namprempre, the results on transitive signatures are joint work with Mihir Bellare.

1.2.1 Identity-based Identification Schemes

Identification schemes. Authentication is a basic need in securing communication. As already pointed out in Section 1.1, we can distinguish between entity

authentication where an entity merely proves to be "alive" at the time of protocol execution, and message authentication that guarantees a received message to be identical to an original message created by a correspondent. Protocols realizing the former are called *identification schemes*, while *message authentication codes* (in the symmetric-key setting) and *signature schemes* (in the public-key setting) are used to realize the latter.

A standard identification (SI) scheme is an interactive protocol to provide entity authentication. Symmetric identification schemes are an essential part of key distribution protocols such as the Kerberos [KN93] and Needham-Schroeder-Lowe [NS78, Low96] protocols, but we will focus on the asymmetric case here. A prover P identifies himself to a verifier V by convincing V that he knows the private key sk corresponding to his public key pk. A trivial solution would be to let the prover simply send the secret key to the verifier, but that would allow an eavesdropper to impersonate the prover after listening in on a single conversation. To circumvent this problem, identification schemes typically involve a challenge-response protocol that does not reveal the secret key itself.

Identification schemes are used in practice to allow so-called *smart cards* (plastic "credit" cards containing a tamper-resistant microchip) to identify as a properly issued card to the card reader. They are also applied in the military as *Identification Friend or Foe* (IFF) systems [MvOV96], to distinguish friendly from hostile aircraft.

Identity-based cryptography. A crucial aspect when implementing public-key cryptography in practice is to provide a secure way of linking users to their public keys. The standard solution is to set up a *public key infrastructure* (PKI), where trusted entities issue *certificates* to assert that a public key belongs to a certain user. A certificate typically contains the identity of the user, the public key, and the trusted entity's signature. Two users wishing to communicate securely first need to exchange certificates or look up each other's certificate in a public directory.

In 1984, Shamir [Sha84] suggested *identity-based cryptography* as a more efficient solution that eliminates the need for user certificates. The idea is to design cryptosystems for which any bit string can be a valid public key, thereby allowing a user's public key to be simply his identity or email address. Of course, the rightful owner of an identity needs some piece of secret information that gives him a computational advantage over other users. He cannot generate this secret by himself, because then an attacker could do the same thing. For this purpose, a trusted *key generation center* is set up to generate a single domain-wide *master public key mpk* and a corresponding *master secret key msk*. The master public key is published as a domain-wide parameter, while the master secret key is kept secret by the key generation center that uses it to compute the *user secret key usk* corresponding to the user's identity. The center is assumed to issue user secret keys over a secure or out-of-band channel.

State of the area prior to this work. Following the work of Fiat and Shamir [FS86], numerous identification schemes based on zero-knowledge protocols have been proposed [FS86, Bet88, CEvdG88, FFS88, MS88, GQ89, Gir90, OO90, OS90, Sch90, Gir91, BM92, Oka93, FF02]. Most of these follow a canonical three-move structure in which the prover starts the protocol with a commitment, the verifier replies with a challenge, and the prover complies by sending an appropriate response. Fiat and Shamir already showed how hash functions can be used to transform such canonical identification schemes into signature schemes, an approach that was later proven to be security-preserving in the random oracle model [OO98, PS00, AABN02].

Many of the above schemes were actually proposed as identity-based identification (IBI) and corresponding identity-based signature (IBS) schemes [FS86, GQ89, Oka93, Gir90, Bet88]. The IBS scheme in Shamir's paper [Sha84] introducing the concept of identity-based cryptography was proposed directly, without an underlying SI or IBI scheme. The introduction of pairings over elliptic curves to cryptography [JN03] caused a new wave of pairing-based IBS schemes being proposed [SOK00, Pat02, Hes03, CC03, Yi03].

When it comes to provable security, however, we found that the above primitives and the relations between them are not fully understood. While a considerable amount of work exists on the provable security of identification schemes, it is limited to standard identification schemes and doesn't take into account the additional risks that are introduced by the multi-user setting of identity-based schemes. The situation for IBS schemes is somewhat better. A definition of security for IBS schemes exists [CC03], and a general transform was even proposed [DKXY03] that under certain conditions turns secure standard signature (SS) schemes into IBS schemes. Several IBS schemes, however, have not yet been proven secure (the most famous example being Shamir's original IBS scheme [Sha84]), either because they cannot be obtained as the application of the general transformation to a secure SS scheme, or because nothing is known about the security of the SS scheme in question.

Our contributions. Our work shines a light in the darkness surrounding the provable security of IBI and IBS schemes by defining a clear security notion for IBI schemes, proving general transformations between some of the notions, and proving the security of known schemes that we revisit, or new schemes that we surface. We briefly summarize the different steps taken here, but we would like to refer the reader to Section 3.1 for a more detailed but still fairly accessible summary of our results in this area.

- As a first step, we extend the security notions of SI schemes under passive, active and concurrent attack to the identity-based setting by additionally allowing the IBI adversary to initiate, interact with and corrupt identities of its choice. Our definition is inspired by security notions of other identity-

based primitives, but formalizing it correctly was not a trivial task due to the interactivity of the problem and the complexity of the attack model.

- We show how to construct a trivial yet inefficient IBI scheme from any SI scheme using certificates, and prove it secure under the new notions.

- We define a class of SI schemes that we call *convertible* SI (cSI) schemes, and show how any such cSI scheme can be transformed into a corresponding IBI scheme while preserving provable security. This transform will be our main tool in analyzing the security of IBI schemes. We observe that a similar transform turning convertible SS (cSS) schemes into IBS schemes is a generalization of the transform of Dodis et al. [DKXY03].

- We also observe that the Fiat-Shamir transform [FS86] turning canonical SI schemes into SS schemes does not generally extrapolate to the identity-based case, and give a counterexample supporting this observation. We present a modified transformation that does guarantee preservation of security when applied to general IBI schemes.

- The main technical part of our work lies in going through a legacy of two decades of proposed SI, IBI and IBS schemes and proving their security based on our general transformation, surfacing previously undefined cSI schemes where necessary, or in one case [Gir90, SSN98] showing an attack when we found the scheme to be insecure. Much to our surprise, we also found that almost all non-trivial IBS schemes known today – including the famous one by Shamir [Sha84] – can be seen as the result of applying our transforms to an appropriate cSI scheme. Therefore we believe that, apart from reducing the burden of proving the security of IBS schemes to that of SI schemes (which appears to be a much easier task indeed), our framework captures a more general idea of how IBI and IBS schemes are constructed, and helps in unifying our view of the area.

- Lastly, we consider the sole exception we found in the literature of a secure IBI scheme that cannot be seen as the transformation of a corresponding cSI scheme. We prove the security of this scheme directly as an IBI scheme (a result that was missing in the original work [Oka93]), and propose a slightly more natural and efficient variant that finds itself in the same situation of needing direct proof. We also surface corresponding IBS schemes through our modified Fiat-Shamir transform.

A schematic overview summarizing our security results for different schemes is given in Figure 3.2 on page 29.

1.2.2 Transitive Signature Schemes

THE CONCEPT. A standard signature scheme allows a signer with public key *spk* to authenticate a message by creating a signature σ with the corresponding secret key *ssk*. The verifier can then later check the validity of the signature using the public key. For security, one would expect signatures to be *unforgeable*, meaning that for an adversary who doesn't know the secret key it is infeasible to create a valid signature himself. The most common security notion for SS schemes is existential unforgeability under chosen-message attack (uf-cma) [GMR88], which says that even after seeing signatures for any (reasonable) number of messages of his choice, the adversary cannot forge a signature for a new message that was not signed before.

The concept of *transitive signature* (TS) schemes as introduced by Micali and Rivest [MR02b] is closely related, but instead of signing arbitrary messages, the signer authenticates edges in a dynamically growing graph. (This work concentrates on undirected graphs; the directed case is still an open problem that may be very hard to solve [Hoh03].) With his secret key *tsk*, the signer can at any time issue a signature for an edge $\{i, j\}$, and thereby acknowledge the existence of the edge $\{i, j\}$ in the graph. Signatures are verified using the corresponding public key *tpk*, but the transitivity additionally requires that any user (so not only the signer) knowing the public key *tpk*, and having signatures σ_1, σ_2 for two adjacent edges $\{i, j\}$ and $\{j, k\}$, is able to compute a third signature σ_3 connecting nodes i and k directly. Through this property, the graph being authenticated does not only contain the edges explicitly signed by the signer, but is the entire transitive closure of this graph. Requiring signatures to be unforgeable in the same way as for SS schemes is unrealistic, since the composition algorithm explicitly allows forgeries to a limited extent. The new security definition [MR02b] however requires that these are the *only* signatures that can be created by an adversary, and that it is impossible to create a signature for an edge that is not in the transitive closure of previously seen signatures.

A TS scheme is trivially realized by letting the verification algorithm of an SS scheme accept a chain of signatures describing a path between two nodes as a valid signature for an edge connecting them directly. Apart from having the disadvantage of an increased signature size, this solution may also be undesirable because of the loss of privacy: a signature reveals its creation history, instead of only certifying the existence of an edge.

Micali and Rivest [MR02b] mention military chains-of-command as a practical application for the directed case, where an edge from person i to person j means that i is a superior of j, and mention proofs of equality of administrative domains (an undirected edge between i and j meaning that they are in the same administrative domain) as an application for the undirected case. A truly compelling application, however, is yet to be found.

Our contributions. Apart from introducing the concept of transitive signatures, Micali and Rivest [MR02b] also presented a first non-trivial and provably secure construction realizing it based on discrete logarithms, that we will refer to as the $\mathcal{DL}$-$\mathcal{TS}$ scheme here. Briefly, we present several new schemes based on various alternative assumptions, and we introduce a hash-based technique that significantly reduces the size of a signature. Finally, we also solve a subtle issue with the correctness definition for TS schemes of Micali and Rivest [MR02b]. More specifically, our contributions are the following. (We refer to Section 2.3 for an explanation of the mathematical assumptions used.)

- Our starting point is an open question raised by Micali and Rivest concerning the security of an RSA-based TS scheme, that we will refer to as $\mathcal{RSA}$-$\mathcal{TS}$ here. They noted that the scheme was only secure against non-adaptive adversaries (i.e. adversaries that have to commit to their signature queries before seeing any signatures). While it is still an open question whether the scheme is secure under adaptive attack assuming the one-wayness of the RSA function, we provide a proof under the stronger one-more RSA assumption.

- We then proceed to answer the natural question if there exists a TS scheme that is secure under adaptive attack assuming only the one-wayness of RSA, by presenting the $\mathcal{Fact}$-$\mathcal{TS}$ scheme that is provably secure under the even weaker factoring assumption.

- We present the $\mathcal{DL}1m$-$\mathcal{TS}$ scheme, which is a more natural variant of the $\mathcal{DL}$-$\mathcal{TS}$ scheme that we prove secure under the one-more discrete logarithm assumption. We also present the $\mathcal{Gap}$-$\mathcal{TS}$ scheme based on gap Diffie-Hellman groups.

- Subsequently, we introduce a technique using hash functions to eliminate the need for so-called *node certificates* in the $\mathcal{RSA}$-$\mathcal{TS}$, $\mathcal{Fact}$-$\mathcal{TS}$ and $\mathcal{Gap}$-$\mathcal{TS}$ schemes, significantly reducing signature sizes for these schemes. The security of the resulting $\mathcal{RSAH}$-$\mathcal{TS}$, $\mathcal{FactH}$-$\mathcal{TS}$ and $\mathcal{GapH}$-$\mathcal{TS}$ relies on the random oracle model, however. We also present a general construction that encompasses all three of these schemes in a single proof.

- Finally, we also address a subtlety that causes the correctness and security definitions of TS schemes to be entangled with each other, and provide a new correctness definition that doesn't have this problem.

We refer to Section 4.1 for a more detailed summary of our results, and in particular to Figures 4.1 and 4.2 for a schematic overview of the cost and security properties associated to all schemes.

1.3 Overview

The remainder of this thesis is structured as follows. In Chapter 2, we first introduce the notation that will be used throughout this text. We further explain the concept of practice-oriented provable security [Bel98], and provide more details on the discussion surrounding the random oracle model. Then we will briefly describe the number-theoretic problems on which the schemes treated in this work are based.

Chapter 3 contains our results related to security proofs for identity-based identification and signature schemes. Its first section contains a more technical summary of our results than presented here in the introduction, and is highly recommended to the technically interested but time-restricted reader (aren't we all...). The following section formalizes security notions for SI, IBI, SS and IBS schemes. The natural certificate-based IBI scheme is described in Section 3.3. After introducing cSI schemes and proving our transforms in Section 3.4, we proceed with the discussion of the schemes that fit our framework in Section 3.5. In Section 3.6, we present the only two IBI schemes that do not originate from a cSI scheme, and we prove their security as an IBI scheme directly.

Our results for transitive signature schemes are presented in Chapter 4. The first section again contains a technical summary of the contributions. Formal definitions for transitive signatures and their security are given in Section 4.2. The starting point of our work, the security proof of the $\mathcal{RSA}$-$\mathcal{TS}$ scheme, is presented in Section 4.4, Section 4.5 revisits to some definitional issues, and Section 4.6 describes new schemes employing the node certification paradigm. We explain how to avoid this paradigm in Section 4.7.

Chapter 5 concludes the thesis with a brief summary of our contributions, and points out a few interesting open problems.

Chapter 2

Preliminaries

2.1 Notation

Let $\{0,1\}$ be the set of individual bits and let $\{0,1\}^*$ be the set of all bit strings. We let $\mathbb{N} = \{0,1,2,\ldots\}$ denote the set of natural numbers. If $k \in \mathbb{N}$, then 1^k is the bit string of k ones and $\{0,1\}^k$ is the set of bit strings of length k. The empty string is denoted ε. If x, y are strings, then $|x|$ is the length of x and $x\|y$ is the concatenation of x and y. If S is a set, then $|S|$ is its cardinality. By the notation $x \xleftarrow{R} S$, we mean that an element x is selected uniformly at random from S. A function $f : \mathbb{N} \to [0,1]$ is said to be *negligible* if it approaches zero faster than the inverse of any polynomial, i.e. for every exponent $c \in \mathbb{N}$ there exists an integer $k_c \in \mathbb{N}$ such that $f(k) \leq k^{-c}$ for all $k > k_c$. The function f is said to be *overwhelming* if its distance to unity approaches zero faster than the inverse of any polynomial, i.e. for every exponent $c \in \mathbb{N}$ there exists an integer $k_c \in \mathbb{N}$ such that $f(k) \geq 1 - k^{-c}$ for all $k > k_c$.

If A is a deterministic algorithm (Turing machine) with access to oracles $\textsc{Or}_1, \ldots, \textsc{Or}_m$, then the notation $y \leftarrow \mathsf{A}(x_1, \ldots, x_n : \textsc{Or}_1, \ldots, \textsc{Or}_m)$ denotes that y is assigned the outcome of running A on inputs $x_1, \ldots, x_n$. A *randomized* or *probabilistic* algorithm A expects a random bit string of length ρ_{A} as an extra input, called the *random tape* of A. We use $y \xleftarrow{R} \mathsf{A}(x_1, \ldots, x_n : \textsc{Or}_1, \ldots, \textsc{Or}_m)$ as a shorthand notation for running A on inputs $x_1, \ldots, x_n, R$ where R is a fresh random tape $R \xleftarrow{R} \{0,1\}^{\rho_{\mathsf{A}}}$ and assigning the outcome to y. We write the set of all possible outputs y of this experiment as $[\mathsf{A}(x_1, \ldots, x_n : \textsc{Or}_1, \ldots, \textsc{Or}_m)]$. All algorithms in this thesis are assumed to be probabilistic, unless otherwise noted.

The running time of an algorithm A is denoted by $\mathbf{T}_{\mathsf{A}}$, and the number of times A queried the $\textsc{Or}_i$ oracle by $\mathbf{Q}_{\mathsf{A}}^{\textsc{Or}_i}$. We define $\mathbf{Q}_{\mathsf{A}} = \sum_i \mathbf{Q}_{\mathsf{A}}^{\textsc{Or}_i}$. For randomized algorithms, we use the expected values of these quantities. Algorithm

A is said to be *polynomial-time* if its running time is bounded by a polynomial in its input length, meaning that there exists a polynomial $p(k)$ such that $\mathbf{T}_{\mathsf{A}} < p(\sum_i |x_i|)$. We assume that an oracle query consumes one time unit, so that the number of oracle queries of a polynomial-time algorithm is also bounded by a polynomial in the input length. Note that the running time of a randomized algorithm may be non-deterministic, but it has to be upper bounded by a polynomial in order to be a probabilistic polynomial-time algorithm.

An interactive algorithm (modelling a party such as prover or verifier in a protocol) is a stateful algorithm that on input an incoming message M_{in} (this is ε if the party is initiating the protocol) and state information St outputs an outgoing message M_{out} and updated state St'. For an interactive algorithm A with access to oracles $\textsc{Or}_1, \ldots, \textsc{Or}_m$, this is written as $(M_{out}, St') \xleftarrow{R} \mathsf{A}(M_{in}, St : \textsc{Or}_1, \ldots, \textsc{Or}_m)$. An interactive algorithm may optionally take a random tape of length ρ_{A} as a third input. This random tape remains the same throughout subsequent invocations, A is supposed to include a pointer to the first unused bit of the random tape in its state information. We use the same shorthand notation as for non-interactive algorithms, omitting the random tape whenever not strictly needed.

2.2 Practice-Oriented Provable Security

Security notions. Before anything useful can be said about the security of a scheme, it must be made perfectly clear what is understood under its security. This is captured by the *security notion* associated to the primitive, which is usually described in terms of a game or experiment. The adversary, modelled as a probabilistic algorithm, is given certain inputs and has access to certain oracles, and is challenged to create an output that is considered "offending" to the scheme.

Deciding on a suitable security notion that is strong enough to be meaningful for practical purposes, yet weak enough to be achievable by real-world schemes, is an important and time-consuming task that should not be underestimated. Security notions may take a long time to crystallize, and in fact it is not uncommon for multiple notions to exist (and have good reason for their existence) in parallel. Public-key encryption for example has an extended history of suggested adversarial goals and attack models, ranging from indistinguishability of ciphertexts under chosen-plaintext attack [GM84], non-adaptive chosen-ciphertext attack [NY90] and adaptive chosen-ciphertext attack; over non-malleability of ciphertexts [DDN91] under the same sorts of attack [DDN91, DDN95, DDN00]; to the very high-level concept of plaintext awareness [BR98]. After the publication of relations among these notions [BDPR98], the community seemed to settle with indistinguishability under adaptive chosen-ciphertext attack as the

"holy grail" for public-key encryption, although Canetti et al. [CKN03] recently suggested to slightly loosen the definition again.

But even if debate exists which notion is best-suited to capture the security of a particular scheme, a security proof under any notion at least forces cryptographers to prove that the proposed scheme meets a certain clearly defined goal. A scheme proven secure under a weaker notion but offering other advantages (e.g. efficiency, patent issues, ...) may still find applications in settings where the stronger security notion is not needed.

Polynomial security. Only few practical cryptographic schemes are *unconditionally* (or *information-theoretically*) secure in the sense that they resist any type of adversary, regardless of its computational power, memory space and other resources. More commonly, we will have to restrict the class of adversaries to polynomial-time probabilistic algorithms, and define security through an asymptotical definition involving a *security parameter* $k \in \mathbb{N}$, which is usually a key length of some sort. Typically, we will let $\mathbf{Exp}^{\text{sec}}_{\mathcal{S},\mathsf{A}}(k)$ denote the outcome (0 if the adversary looses, 1 if it wins) of a single run of the security experiment associated to security notion sec confronting adversary A with scheme $\mathcal{S}$ using security parameter k. The *advantage* of A is the probability that it wins the game:

$$\mathbf{Adv}^{\text{sec}}_{\mathcal{S},\mathsf{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{sec}}_{\mathcal{S},\mathsf{A}}(k) = 1\right],$$

where the probability is taken over the coins tossed by A and by the experiment. Scheme $\mathcal{S}$ is finally said to be (polynomially) secure under notion sec if the advantage of A is a negligible function in k for all polynomial-time adversaries A.

Concrete security. A common critique on provable security often heard from practitioners is that asymptotical bounds don't provide the level of detail that is desired in practice. Polynomial security guarantees that the scheme will be secure for "sufficiently large" values of the security parameter, without suggesting any concrete values for it. What practitioners want, however, are raw numbers telling them exactly how long an encryption key should be to make sure that an adversary with 2^{80} time steps and 2^{40} decryption queries to spend on an attack has at most a chance of 2^{-20} to break the scheme. Such quantitative information is only vaguely represented in asymptotical security statements that involve polynomials and negligible functions.

A more precise approach is *practice-oriented provable security*, which obtains concrete bounds by investigating the details of the reduction algorithm presented in the security proof. Indeed, the description of the algorithm reveals *how much* easier or harder it is to break the scheme than to break one of the underlying building blocks, and this information can be made explicit by stating attributes such as running time, number of oracle queries and advantage of the reduction algorithm as part of the security claim. As a consequence, security is

no longer a binary property (secure or insecure), but a continuous one: between two polynomially secure schemes, one can be *more secure* than the other. If for example scheme $\mathcal{S}_1$ has advantage 2^{-k} while scheme $\mathcal{S}_2$ has advantage $2^{-k/2}$, then $\mathcal{S}_2$ needs twice the keylength of scheme $\mathcal{S}_1$ to achieve the same security level. We say that $\mathcal{S}_1$ has a *tighter* security reduction than $\mathcal{S}_2$ and hence may be preferable over $\mathcal{S}_2$.

THE RANDOM ORACLE MODEL. Some cryptographic primitives seem particularly ill-suited for provable security, in the sense that none of the proposed provably secure constructions is efficient enough to be used in practice, causing practitioners to turn to heuristic but more efficient alternatives instead. In the mid-1990s, public-key encryption and signature schemes were notorious examples of such primitives. Bellare and Rogaway [BR93a] suggested the *random oracle model* as a compromise between theory and practice, sacrificing a piece of provable security to buy efficiency. They subsequently used this paradigm to prove the security of their OAEP (Optimal Asymmetric Encryption Padding) [BR98, Sho02] and PSS (Probabilistic Signature Scheme) schemes [BR96]. These are just as efficient as the heuristic schemes in use at the time, and gradually made their way into industrial standards such as PKCS#1 and IEEE P1363. The random oracle model has since been used in numerous security proofs throughout the literature (see for example [BF01, BLS01, PS00, AABN02, Bol03a]).

The idea of the random oracle model is to prove security in an imaginary model where all algorithms, including the adversary, have access to an oracle H that implements a random function, meaning that images are independently and uniformly distributed over the domain of the function. When implemented in practice, the random oracle is replaced with a "good" cryptographic hash function such as SHA-1 [Nat95] or RIPEMD-160 [DBP96]. The hash functions are hoped to sufficiently mimic the unpredictable behavior of a random oracle to preserve security in the real world.

A lot of controversy exists about the true value of security proofs in the random oracle model, and indeed, they should be treated with care. Assuming hash functions to behave like random oracles is not just a strong assumption, it is plainly false: obviously, no efficiently computable function can ever be assumed to be unpredictable. Moreover, critiques have been found separating the random oracle model from the standard model. Canetti, Goldreich and Halevi [CGH98] designed (contrived) encryption and signature schemes that are secure in the random oracle model, but that are insecure when the random oracle is instantiated with *any* function ensemble. Later, Nielsen [Nie02] showed that the (quite natural) problem of non-interactive non-committing encryption has no solution in the standard model, while it has a simple solution in the random oracle model. Lastly, Goldwasser and Tauman [GK03] demonstrated the existence of a (contrived) identification scheme for which the Fiat-Shamir transform produces an insecure signature scheme when the random oracle is instantiated with

any function ensemble. The Fiat-Shamir transform [FS86] (referred to as the fs-I-2-S transform in Construction 3.3 of this thesis) converts a class of identification schemes into signature schemes and was proven to be security-preserving in the random oracle model [PS00, AABN02].

Hence, from a theoretical point of view, a proof in the random oracle model is unmistakably inferior to a proof in the standard model. In practice, however, attacks on schemes involving hash functions often assume the output of the hash function to be random themselves. Proofs in the random oracle models do guard against such attacks. Moreover, a scheme proven to achieve some clearly specified security notion in the random oracle model is still strongly preferable over completely ad-hoc protocol design. We advocate that the random oracle model is to be judged on its merits: it provided "provable" security for schemes that withstood years of attacks but lacked a proof in the standard model, and it has proven to be a powerful tool for the design and analysis of new efficient cryptographic protocols.

2.3 Mathematical Assumptions

The security of a scheme can be proved based on the security of its underlying primitives, or can be reduced directly from a supposedly hard mathematical problem. In this section, we describe the mathematical assumptions that will be used throughout the rest of this work.

2.3.1 Factoring and RSA

The factoring problem. Despite being probably one of the most widely studied problems in number theory, no polynomial-time algorithm is known to compute the prime factors of general integers. Because efficient algorithms do exist for integers with small prime factors, the composites used for cryptography are mostly products of two large primes. We define a *modulus generator* K_{fact} as a polynomial-time algorithm that, on input 1^k, outputs a modulus N together with primes p, q such that $N = pq$. The factoring problem associated to K_{fact} is formally defined through the following experiment:

Experiment $\mathbf{Exp}^{\text{fact}}_{\mathsf{K}_{\text{fact}},\mathsf{A}}(k)$
$(N, p, q) \xleftarrow{R} \mathsf{K}_{\text{fact}}(1^k)$
$r \xleftarrow{R} \mathsf{A}(N)$
If $1 < r < N$ and $\gcd(r, N) > 1$ then return 1 else return 0.

The factoring advantage of algorithm A is

$$\mathbf{Adv}^{\text{fact}}_{\mathsf{K}_{\text{fact}},\mathsf{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{fact}}_{\mathsf{K}_{\text{fact}},\mathsf{A}}(k) = 1\right]$$

and the factoring problem associated to K_{fact} is said to be hard if this advantage is a negligible function of k for all algorithms A with running time polynomial in k. The hardness of the factoring problem is mainly impacted by the length of the prime factors p, q (and hence of N). The fastest factoring algorithm for equal-length factors $|p| = |q|$ known to date is the Number Field Sieve and has running time $O(e^{1.923|N|^{1/3}\ln(|N|)^{2/3}})$ [LL93]. The modulus typically should be about 1024 bits long to be secure against adversaries spending 2^{80} time steps. Such a modulus was recently estimated to be factorable within one year using special hardware costing about 10 million US$ [ST03].

The set $\mathbb{Z}_N^* = \{1 \leq x < N \ : \ \gcd(x, N) = 1\}$ forms a group with the multiplication modulo N. Proofs reducing the security of a scheme from the factoring problem often exploit the fact that two square roots x_1 and x_2 of the same element $y \in \mathbb{Z}_N^*$ reveal the factorization of N if $x_1 \not\equiv \pm x_2 \bmod N$. Indeed, $x_1^2 \equiv x_2^2 \bmod N$ implies that $(x_1 - x_2)(x_1 + x_2) = 0 \bmod N$ and hence that $(x_1 - x_2)(x_1 + x_2)$ is an integer multiple of N, while neither $(x_1 - x_2)$ nor $(x_1 + x_2)$ are multiples of N because of the condition that $x_1 \not\equiv \pm x_2 \bmod N$. A factor of N can then be computed as $\gcd(x_1 - x_2, N)$.

If p is an odd prime and a is an integer, then the *Legendre symbol* of a with respect to p is defined as

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a \equiv 0 \bmod p \\ +1 & \text{if } a \text{ is a square mod} p \\ -1 & \text{if } a \text{ is a non-square mod} p \end{cases}$$

and can be efficiently computed as

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \bmod p \ .$$

For a composite modulus $N = p_1 \cdot \ldots \cdot p_k$ with $p_1, \ldots, p_k$ prime, the *Jacobi symbol* of a with respect to N is defined as

$$\left(\frac{a}{N}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right) \ .$$

From the above definition, it might seem that the Jacobi symbol can only be efficiently computed if the factorization of N is known. Using special properties of the Jacobi symbol however, it can also be computed when the factorization of the modulus is not known (see e.g. [MvOV96]).

A Blum integer is an integer $N = pq$ where p, q are primes such that $p \equiv q \equiv 3 \bmod 4$. Blum integers have the special property that -1 is a non-square with Jacobi symbol $+1$ modulo N. A *Blum modulus generator* K_{blum} is a modulus generator that only generates Blum integers N.

THE RSA PROBLEM. The RSA problem, named after its inventors Rivest, Shamir and Adleman [RSA78], is related to the factoring problem but is not known to be equivalent with it. An *RSA key generator* K_{rsa} is a randomized polynomial-time algorithm that on input 1^k returns a modulus N, an encryption exponent e and a decryption exponent d such that N is the product of two large primes and $ed \equiv 1 \bmod \varphi(N)$, where $\varphi(N) = (p-1)(q-1)$ is the Euler totient function. Since $\varphi(N)$ is also the order of the group $\mathbb{Z}_N^*$, the latter property ensures that $(x^e)^d \equiv x \bmod N$ for all $x \in \mathbb{Z}_N^*$. The RSA problem associated to K_{rsa} is defined through the following experiment:

Experiment $\mathbf{Exp}^{\text{rsa}}_{\mathsf{K}_{\text{rsa}},\mathsf{A}}(k)$
$(N, e, d) \xleftarrow{R} \mathsf{K}_{\text{rsa}}(1^k)$
$y \xleftarrow{R} \mathbb{Z}_N^*$
$x \xleftarrow{R} \mathsf{A}(N, e, y)$
If $x^e \equiv y \bmod N$ then return 1 else return 0.

The advantage of algorithm A in inverting RSA is

$$\mathbf{Adv}^{\text{rsa}}_{\mathsf{K}_{\text{rsa}},\mathsf{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{rsa}}_{\mathsf{K}_{\text{rsa}},\mathsf{A}}(k) = 1\right],$$

and the RSA one-wayness assumption associated to K_{rsa} says that the above is a negligible quantity in k for all polynomial-time algorithms A.

Given the factorization of N, it is easy to compute the group order $\varphi(N)$ and hence to solve the RSA problem by computing d as the inverse of e modulo $\varphi(N)$. Conversely, it is also known that given an encryption exponent e and the corresponding decryption exponent d, computing the factorization of N becomes easy [Bon99]. It is not known however whether there exists a way to solve the RSA inversion problem without requiring to factor the modulus. This implies that the RSA assumption is at least as strong as the factoring assumption. (The commonly used terminology for assumptions can be confusing, in that a *weak* assumption is preferable over a *strong* one. It would be clearer to speak about *light* and *heavy* assumptions instead, but viewing its widespread adoption we continue to use the conventional terminology in this work.)

Note that we don't make any assumptions on the distribution of moduli and exponents generated by K_{rsa}, but only assume that their associated RSA problem is hard. In particular, some schemes may use small encryption exponents (e.g. $e = 3$) for efficient encryption, others may need e to be prime for security reasons. These are all captured under our definition of RSA key generators.

THE ONE-MORE RSA PROBLEM. As early as in 1982, Chaum proposed a blind signature scheme [Cha83] exploiting the multiplicative homomorphism of the RSA function. For almost two decades, the security of this scheme remained unclear: while no proof for it was known, neither did anyone propose a successful

attack. Bellare et al. suggested that the reason behind this situation might be that the security relies on properties of the RSA function that go beyond mere one-wayness. For this purpose, they defined the stronger *one-more RSA assumption* and proved Chaum's blind signature scheme secure under it. The same assumption was used later to prove the GQ identification scheme secure under active and concurrent attack [BP02].

The one-more RSA problem gives the adversary access to two oracles: a challenge oracle CHALL that on each invocation returns a new target point chosen uniformly at random from $\mathbb{Z}_N^*$, and an inversion oracle INV$(\cdot)$ that on input $y \in \mathbb{Z}_N^*$ returns $x \in \mathbb{Z}_N^*$ such that $x^e \equiv y \bmod N$. The adversary's task is to invert all target points generated by the challenge oracle using strictly less inversion queries than the total number of target points inverted. The game is described more formally be the following experiment:

Experiment $\mathbf{Exp}^{\text{1m-rsa}}_{\mathsf{K}_{\text{rsa}},\mathsf{A}}(k)$
$(N, e, d) \xleftarrow{R} \mathsf{K}_{\text{rsa}}(1^k)$
$(x_1, \ldots, x_n) \leftarrow \mathsf{A}(N, e : \text{INV}, \text{CHALL})$ where n is the number of queries made to CHALL
Let m be the number of queries made to INV
Let $y_1, \ldots, y_n$ be the challenges returned by CHALL
If $m < n$ and $\forall\, i \in \{1, \ldots, n\} : x_i^e \equiv y_i \bmod N$
then return 1 else return 0.

The advantage of algorithm A is defined as

$$\mathbf{Adv}^{\text{1m-rsa}}_{\mathsf{K}_{\text{rsa}},\mathsf{A}}(k) = \Pr[\mathbf{Exp}^{\text{1m-rsa}}_{\mathsf{K}_{\text{rsa}},\mathsf{A}}(k) = 1].$$

The one-more RSA assumption says that this is a negligible function for all adversaries A with running time polynomial in k.

Viewing the novelty of the assumption, it should be used with caution, since it did not receive as much scrutiny from experts in the field yet as the standard RSA problem. Also note that the one-more RSA assumption only holds if factoring does not reduce to the RSA problem: if the ability to compute RSA inversions enables one to factor the modulus, then the one-more RSA problem is easily solvable.

2.3.2 Discrete Logarithms

A *discrete logarithm group generator* K_{dlog} is a randomized algorithm that on input 1^k generates a triplet $(\mathbb{G}, q, g)$, where $\mathbb{G}$ is the compact description of a multiplicative cyclic group of order q, and g is a generator of $\mathbb{G}$. We will use the notation $\mathbb{G}$ interchangeably for the group itself and its compact description. We assume that the group operation $\cdot$ and the operator $\equiv$ testing element equality are efficiently computable from the description of the group.

The discrete logarithm of $y \in \mathbb{G}$ with respect to g is $x = \log_g(y) \in \mathbb{Z}_q$ such that $y \equiv g^x$. The discrete logarithm problem is to compute the discrete logarithm of uniformly distributed elements of $\mathbb{G}$ with respect to g:

Experiment $\mathbf{Exp}^{\text{dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k)$
$(\mathbb{G}, q, g) \xleftarrow{R} \mathsf{K}_{\text{dlog}}(1^k)$
$x \xleftarrow{R} \mathbb{Z}_q$; $y \leftarrow g^x$
$x' \xleftarrow{R} \mathsf{A}(\mathbb{G}, q, g, y)$
If $g^{x'} \equiv y$ then return 1 else return 0.

and the advantage of A is its probability of winning this game:

$$\mathbf{Adv}^{\text{dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k) = 1\right].$$

The discrete logarithm problem associated to K_{dlog} is hard if $\mathbf{Adv}^{\text{dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k)$ is a negligible function in k for all polynomial-time algorithms A.

Different types of discrete logarithm groups are in use today. The best-known example is the multiplicative group $\mathbb{Z}_p^*$ of order $q = p-1$ with p prime. The size of p determines the hardness of the discrete logarithm problem; the fastest known algorithm is the index calculus [Adl79] and has complexity $O(e^{c\sqrt{\ln q \cdot \ln\ln q}})$. Roughly speaking, p should have the same length as an RSA modulus to obtain the same level of security, say 1024 bits. An idea of Schnorr [Sch90] was to use a smaller subgroup of $\mathbb{Z}_p^*$ of prime order $q \,|\, p-1$. The size of q can then be reduced to about 160 bits without affecting the hardness of the problem. Note that the representation of a group element still occupies $|p| \approx 1024$ bits. Elliptic curve groups over finite fields are a third type of groups where discrete logarithms are assumed to be hard. While having the disadvantage of being introduced to cryptography more recently [Mil86, Kob87] and hence possibly being more vulnerable to mathematical breakthroughs, they have the major advantage that the representation of an element is only slightly longer than the prime order q, or about 170 bits long to achieve comparable security to a 1024-bit RSA modulus. No subexponential discrete logarithm algorithm is known in elliptic curve groups; the fastest is the Pollard-rho method [Pol78] which has complexity $O(\sqrt{q})$. Comparing key sizes between different cryptographic problems is a difficult task, and depends heavily on the assumptions made about algorithmic advances in the near future. We refer to Lenstra and Verheul [LV01] for an extensive overview.

Analogously to the one-more RSA problem, Bellare et al. [BNPS03] also defined the one-more discrete logarithm problem in which an adversary is given access to a challenge oracle CHALL generating a new random target point every time it is called, and a discrete logarithm oracle DLOG that on input $y \in \mathbb{G}$ returns $x \in \mathbb{Z}_q$ such that $g^x \equiv y$. The fact that no efficient algorithm is known to implement such oracle does not pose a problem, since oracles are not restricted

to polynomial time. The adversary's task is to compute the discrete logarithm of all target points using strictly less DLog queries.

Experiment $\mathbf{Exp}^{\text{1m-dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{A},(}k)$:
$(\mathbb{G}, q, g) \xleftarrow{R} \mathsf{K}_{\text{dlog}}(1^k)$
$(x_1, \ldots, x_n) \leftarrow \mathsf{A}(\mathbb{G}, q, g : \text{DLog}, \text{Chall})$ where n is the number of queries made to Chall
Let m be the number of queries made to DLog
Let $y_1, \ldots, y_n$ be the challenges returned by Chall
If $m < n$ and $\forall\, i \in \{1, \ldots, n\} : g^{x_i} \equiv y_i$
then return 1 else return 0.

The advantage of A is defined as

$$\mathbf{Adv}^{\text{1m-dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{1m-dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k) = 1\right].$$

The one-more discrete logarithm assumption says that the above advantage is a negligible function for all polynomial-time adversaries A. This assumption was used to prove the security of the Schnorr identification scheme [Sch90] under active and concurrent attack [BP02].

The Diffie-Hellman problems. A problem related to the discrete logarithm problem is the computational Diffie-Hellman (CDH) problem, which is at the basis of the Diffie-Hellman key exchange protocol [DH76] and asks to compute g^{ab} given g^a and g^b:

Experiment $\mathbf{Exp}^{\text{cdh}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k)$
$(\mathbb{G}, q, g) \xleftarrow{R} \mathsf{K}_{\text{dlog}}(1^k)$
$a \xleftarrow{R} \mathbb{Z}_q$; $b \xleftarrow{R} \mathbb{Z}_q$; $u \leftarrow g^a$; $v \leftarrow g^b$
$w \xleftarrow{R} \mathsf{A}(\mathbb{G}, q, g, u, v)$
If $w \equiv g^{ab}$ then return 1 else return 0.

The advantage of A, as usual, is defined as the probability of A winning the above game.

The decisional Diffie-Hellman (DDH) problem is to differentiate between Diffie-Hellman tuples and random tuples. Because any adversary can obtain a success probability of 1/2 by random guessing, we parameterize the experiment with a bit d:

Experiment $\mathbf{Exp}^{\text{ddh-}d}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k)$
$(\mathbb{G}, q, g) \xleftarrow{R} \mathsf{K}_{\text{dlog}}(1^k)$
$a \xleftarrow{R} \mathbb{Z}_q$; $b \xleftarrow{R} \mathbb{Z}_q$; $u \leftarrow g^a$; $v \leftarrow g^b$
If $d = 0$ then $w \leftarrow g^{ab}$ else $w \xleftarrow{R} \mathbb{G}$
$d' \xleftarrow{R} \mathsf{A}(\mathbb{G}, q, g, u, v, w)$
If $d' = d$ then return 1 else return 0.

and define the advantage function as A's ability to distinguish between the experiments for $d = 0$ and $d = 1$:

$$\mathbf{Adv}^{\text{ddh}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k) = \left|\Pr\left[\mathbf{Exp}^{\text{ddh-0}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ddh-1}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k) = 1\right]\right|.$$

The hardness of the CDH and DDH problems says that the respective advantage functions are negligible in k for any polynomial-time algorithm A. It is clear that if the CDH problem is easy then so is the DDH problem, and that if computing discrete logarithms is easy then both Diffie-Hellman problems are easy as well. The converses however are not known to hold. Moreover, Joux and Nguyen [JN03] provided evidence for a possible separation (or gap) between the CDH and DDH problems by showing a group in which the DDH problem is easy, while the CDH is still assumed to be hard. We call such groups *Gap Diffie-Hellman (Gap-DH) groups*, and define a *Gap Diffie-Hellman group specifier* as a pair of algorithms $(\mathsf{K}_{\text{gap}}, \mathsf{S}_{\text{ddh}})$, where K_{gap} is a discrete logarithm group generator, and S_{ddh} is a DDH algorithm that on input $\hat{\mathbb{G}}, q, g, g^a, g^b, g^c$ decides whether $c \equiv ab \bmod q$ or not for all $(\hat{\mathbb{G}}, q, g) \in [\mathsf{K}_{\text{gap}}(1^k)]$ and for all $k \in \mathbb{N}$. The *Gap-DH assumption* associated to K_{gap} says that the CDH problem in $\hat{\mathbb{G}}$ is hard, even though the DDH problem is easy. The Gap-DH assumption was originally used by Boneh, Lynn and Shacham [BLS01] to construct a signature scheme with very short signatures, and later by Boldyreva [Bol03a] to construct threshold and multi-signatures.

Similarly to the RSA and the discrete logarithm problem, the CDH problem can be generalized to a one-more variant called the *one-more CDH problem* [Bol03a]. The adversary gets $u \equiv g^a$ as input and has access to a challenge oracle Chall generating random target points $v_i \equiv g^{b_i}$ and a CDH oracle Cdh that on input $v \in \mathbb{G}$ returns $w \equiv v^a$. In order to win the game, the adversary has to output $w_i \equiv g^{ab_i}$ for all target points using strictly less exponentiation queries.

Experiment $\mathbf{Exp}^{\text{1m-cdh}}_{\mathsf{K}_{\text{dlog}},\mathsf{A}}(k)$
  $(\mathbb{G}, q, g) \xleftarrow{R} \mathsf{K}_{\text{dlog}}(1^k)$
  $a \xleftarrow{R} \mathbb{Z}_q$; $u \leftarrow g^a$
  $(w_1, \ldots, w_n) \leftarrow \mathsf{A}(\mathbb{G}, q, g, u : \text{Cdh}, \text{Chall})$ where n is the number of queries made to Chall
  Let m be the number of queries made to Cdh
  Let $v_1 \equiv g^{b_1}, \ldots, v_n \equiv g^{b_n}$ be the challenges returned by Chall
  If $m < n$ and $\forall\, i \in \{1, \ldots, n\} : w_i \equiv g^{ab_i}$
  then return 1 else return 0.

The one-more CDH assumption says that no polynomial-time algorithm A has a non-negligible chance of winning the game, and the *one-more Gap-DH assumption* says that this is still the case when A is additionally given access to

a DDH algorithm. This assumption was previously used by Boldyreva [Bol03a] to prove the security of a blind signature scheme based on Gap-DH groups.

PAIRING FUNCTIONS. The only Gap-DH groups known today are supersingular elliptic curves on which the DDH problem can be solved using the modified Weil or Tate pairing function. The mathematical details of these groups are beyond the scope of this thesis (we refer the interested reader to [JN03]), but some of the schemes treated here need slightly more detail than provided by the black-box definition of Gap-DH groups.

A *pairing generator* is a polynomial-time randomized algorithm K_{pair} that on input 1^k outputs a tuple $(\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e})$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are the compact descriptions of an additive and multiplicative group, respectively, both of the same prime order q. The point P is a generator of $\mathbb{G}_1$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is the *pairing function* with the following properties:

- *Non-degenerate:* $\hat{e}$ does not map all pairs of elements in $\mathbb{G}_1$ to the neutral element of $\mathbb{G}_2$;
- *Computable:* the pairing $\hat{e}(P, Q)$ is computable in polynomial time for all $P, Q \in \mathbb{G}_1$;
- *Bilinear:* $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and for all $a, b \in \mathbb{Z}_q$.

We will use additive notation for $\mathbb{G}_1$ as is conventional for elliptic curve groups. This might be a bit confusing though: the "discrete logarithm" of $Q \in \mathbb{G}_1$ with respect to P is x such that $Q \equiv xP$, the CDH problem is to compute abP given $(\mathbb{G}_1, \mathbb{G}_2, q, P, aP, bP)$, and the DDH problem is to, given $(\mathbb{G}_1, \mathbb{G}_2, q, P, aP, bP, cP)$, decide whether $c \equiv ab \bmod q$. The group $\mathbb{G}_1$ is a Gap-DH group since the DDH problem can be solved by checking that $\hat{e}(P, cP) \equiv \hat{e}(aP, bP)$, while the CDH problem in $\mathbb{G}_1$ is assumed to be hard. The CDH assumption in $\mathbb{G}_1$ is a weaker assumption than the so-called *bilinear CDH assumption* used by Boneh and Franklin [BF01] which states that, given (P, aP, bP, cP), computing $\hat{e}(P, P)^{abc}$ is hard.

Chapter 3

Identity-Based Identification Schemes

3.1 Introduction and Main Contributions

Identity-based cryptography. In 1984, Shamir [Sha84] introduced *identity-based cryptography* as a way to eliminate the public key distribution problem and the associated user certificates. The idea is to design cryptosystems such that any bit string is a valid public key, thereby allowing a user's public key to be simply his identity or email address. A trusted *key generation center* is set up to generate a domain-wide *master public key mpk* and a corresponding *master secret key msk*. The key generation center uses the master secret key to compute the *user secret key usk* corresponding to a user's identity. The center is assumed to issue user secret keys over a secure or out-of-band channel.

While this approach indeed eliminates the need for exchanging or looking up certificates of correspondents, it also has its disadvantages. *Key escrow*, for example, is inherent to identity-based cryptography as described above: the key generation center knows the secret keys of all users in the system, and can hence decrypt messages, forge signatures or impersonate users at will. Users need have unconditional trust in the key generation center, which is undoubtedly unacceptable for many applications. The level of trust can be lowered by using *self-certified* public keys [Gir91], which are generated interactively so that the key generation center does not know the corresponding secret key. This does not prevent the center from creating a new secret key by simulating the interaction, but the user can demonstrate such fraudulent behavior by proving that a secret key different from his own was used.

Also, communication can be done in a fully identity-based manner within

one domain, but classical PKI techniques will still be necessary to make the master public key securely available to users outside the domain boundaries. The number of online lookups will be greatly reduced though, since there will be considerably less key generation centers than total users. Hierarchical identity-based cryptography [HNZI99, GS02] allows for an entire identity-based PKI, but at the cost of a linear growth of the communication complexity with the depth of the hierarchy.

Another problem of identity-based cryptography is key revocation. If a user's key is compromised, he "loses" his identity, because the secret corresponding to it has leaked. A partial solution is to append the current year (or date) to the user's identity [BF01], and use that as a public key. The user then has to retrieve a new secret key from the key generation center every year (day), but the damage of loss of keys remains limited to a single time frame.

The goal of this work is to provide clear security guarantees for identity-based identification and signature schemes. We do not advocate identity-based cryptography as a replacement for traditional PKIs, but rather as an alternative technology with its own advantages and disadvantages. Identity-based cryptography may find practical applications when the efficiency improvements are thought to compensate for the above disadvantages, which may be the case in low-security settings (e.g. user-friendly email encryption to Hotmail accounts) or situations where a trusted superior is readily available (e.g. identification of smart cards issued by a single company).

Current state of the area. The late eighties and early nineties saw the proposal of many identity-based identification (IBI) and identity-based signature (IBS) schemes. These include the Fiat-Shamir IBI and IBS schemes [FS86], the Guillou-Quisquater IBI and IBS schemes [GQ89], the IBS scheme in Shamir's paper [Sha84] introducing identity-based cryptography, and others [Oka93, Gir90, Bet88]. Recently, new pairing-based IBS schemes have been proposed [SOK00, Hes03, Pat02, CC03, Yi03].

Prompted by the renewed interest in identity-based cryptography that has followed the proposal of a practical identity-based encryption (IBE) scheme [BF01], we decided to revisit the IBI and IBS areas. An examination of past work revealed the following.

Although there is a lot of work on proving security in the identification domain, it pertains to standard rather than identity-based schemes. (For example, security proofs have been provided for standard identification schemes related to the Fiat-Shamir and Guillou-Quisquater IBI schemes [FFS88, BP02], but not for the IBI schemes themselves.) In fact, a provable-security treatment of IBI schemes is entirely lacking: there are no security definitions, and none of the existing schemes is proven secure. Given the large number of proposed IBI schemes, this is an important (and quite surprising) gap.

The situation for IBS is somewhat better. Cha and Cheon provide a definition

of security for IBS schemes and prove their scheme secure [CC03]. Dodis, Katz, Xu, and Yung [DKXY03] define a class of standard signature (SS) schemes that they call trapdoor, and then present a random-oracle-using transformation (let us call it tSS-2-IBS) that turns any secure trapdoor SS (tSS) scheme into a secure IBS scheme. Security proofs for several existing IBS schemes, including those of [FS86, GQ89], are obtained by observing that these are the result of applying tSS-2-IBS to underlying tSS schemes already proven secure in the literature [PS00, OO98, AABN02]. However, as we will see, there are several IBS schemes not yet proven secure (one example is Shamir's IBS scheme [Sha84]), either because they are not the result of applying tSS-2-IBS to a tSS scheme, or because, although they are, the tSS scheme in question has not yet been analyzed.

Security notions. The first step, naturally, is defining security notions. We extend to the IBI setting the three notions of security for standard identification (SI) schemes, namely security against impersonation under passive attacks (imp-pa), active attacks (imp-aa) [FFS88], and concurrent attacks (imp-ca) [BP02]. Our model allows the adversary to expose user (prover) keys, and to mount either passive, active, or concurrent attacks on the provers, winning if it succeeds in impersonating a prover of its choice. We remark that although existing security definitions for other identity-based primitives [BF01, CC03, DKXY03] give us some guidance as to which adversary capabilities to consider, there are some issues in the definition for IBI that need thought, mainly related to which capabilities the adversary gets in which stage of its two-stage attack. See Section 3.2.

Certificate-based schemes. Before executing the main task of analyzing practical IBI and IBS schemes, we pause to consider the following natural design of an IBI scheme, based on any given SI scheme, via the certification technique. The authority picks a public and secret key pair (pk, sk) for the SI scheme, and provides these to prover I along with a certificate $cert$ consisting of the authority's signature on I, pk. The prover can now flow $pk, cert$ to the verifier and then identify itself via the SI scheme under pk. The verifier needs to know only I and the public key of the authority in order to authenticate the prover.

Theorem 3.2 says that the above yields a secure IBI scheme. An analogous result holds in the IBS case. We believe that this is worth noting because it highlights the fact that, unlike IBE [BF01], IBI and IBS are trivial to achieve (and in particular do not require random oracles), and enables us to better understand what the practical schemes are trying to achieve, namely to beat the trivial certification-based schemes in performance.

Main contributions and approach. This paper delivers security proofs for a large number of practical IBI and IBS schemes, including not only the ones mentioned above, but many more that we surface as having been, with hindsight,

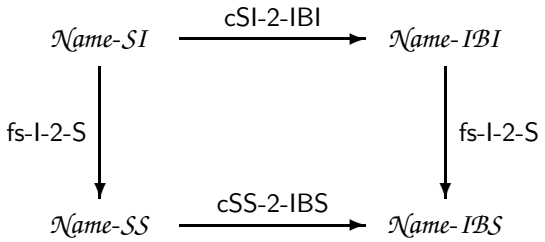


Figure 3.1: Family of schemes associated to a cSI scheme *Name-SI*. If *Name-SI* is imp-atk secure then *Name-IBI* is also imp-atk secure, for all atk $\in \{\text{pa}, \text{aa}, \text{ca}\}$. If *Name-SI* is imp-pa secure then *Name-IBS* is uf-cma secure. Implicit in drawing the diagram this way is that fs-I-2-S(cSI-2-IBI(*Name-SI*)) = cSS-2-IBS(fs-I-2-S(*Name-SI*)).

implicit in the literature.

We do this in two steps. In the first step, we provide a framework that (in most cases) reduces proving security of IBI or IBS schemes to proving security of an underlying SI scheme. In a few cases, we found that the SI schemes in question were already analyzed in the literature, but in many cases they were not. The second step, where lies the main technical contribution of this chapter, is to provide security proofs for those SI schemes not already proven secure, and then provide direct security proofs for the few exceptional IBI or IBS schemes that do not fall under our framework.

The framework, we believe, is of value beyond its ability to reduce proving security of IBI and IBS schemes to proving security of SI schemes. It helps understand how schemes are being derived, and in the process surfaces the implicit schemes we mentioned above. Overall, the framework contributes to simplifying and unifying our picture of the area. We now explain the framework, which is based on a set of transforms, and then summarize the results for specific schemes.

The transforms. We introduce a class of SI schemes that we call *convertible*. The idea is that their key-generation process is based on a primitive called a trapdoor samplable relation that we introduce in Definition 3.5. We then present a transformation cSI-2-IBI that transforms a convertible SI (cSI) scheme into an IBI scheme, and show that it is security-preserving, meaning that if the starting cSI scheme is secure against impersonation under passive, active or concurrent attack, then so is the resulting IBI scheme (in the random oracle model). This will be our main tool for proving security of IBI schemes.

We analogously define convertible standard signature (cSS) schemes and a transform cSS-2-IBS that turns a secure cSS scheme into a secure IBS scheme. These extend the tSS schemes [DKXY03] in the sense that any tSS scheme is also a cSS scheme, and our cSS-2-IBS transform coincides with the tSS-2-IBS transform when the starting scheme is a tSS scheme, but the class of cSS schemes

is larger than the class of tSS schemes.

It is known that the (random-oracle-using) Fiat-Shamir transform [FS86], to which we refer as the fs-I-2-S transform, turns SI schemes into SS schemes. We know that if the former is secure under passive attack, then the latter is unforgeable under chosen-message attack [AABN02]. (Application of the transform and this last result requires that the starting SI scheme is a three-move protocol satisfying a certain technical condition, but these conditions will always be true for the applications we consider.)

Putting the above together yields Corollary 3.11, which says that, as long as a cSI scheme $\mathcal{SI}$ is secure under passive attack, then the IBS scheme $\mathcal{IBS} = \mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{SI}))$ is unforgeable under chosen-message attack. This will be our main tool for proving security of IBS schemes.

We note that fs-I-2-S also transforms a given IBI scheme into an IBS scheme. Furthermore, the diagram of Figure 3.1 commutes, in the sense that $\mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{SI})) = \mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{SI}))$ for any cSI scheme $\mathcal{SI}$.

As an aside, we demonstrate that the analogue of the result of Abdalla et al. [AABN02] does *not* hold in general for fs-I-2-S as a transform of IBI schemes to IBS schemes: Proposition 3.12 shows that there exists an imp-pa secure IBI scheme that when transformed under fs-I-2-S yields an insecure IBS scheme. This does not contradict the above since the scheme in question is not the result of cSI-2-IBI applied to a cSI scheme, but it makes things a little more difficult in a few exceptional cases (that we will discuss later) where we want to derive an IBS scheme from an IBI scheme that is not the cSI-2-IBI transform of any cSI scheme. For this purpose, we slightly modify the fs-I-2-S transform to obtain the efs-IBI-2-IBS transform that turns *any* imp-pa secure IBI scheme into a uf-cma secure IBS scheme.

Scheme families. We seek to explain all IBI schemes $\mathcal{IBI}$ in the literature by surfacing a cSI scheme $\mathcal{SI}$ such that $\mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{SI}) = \mathcal{IBI}$. We seek to explain any IBS scheme $\mathcal{IBS}$ in the literature by surfacing a cSI scheme $\mathcal{SI}$ such that $\mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{SI})) = \mathcal{IBS}$. We are able to do this for most of the schemes we found in the literature [FS86, GQ89, Sha84, Gir90, Hes03, CC03, Yi03, Bet88] including the RSA-based IBI scheme in [Oka93], which reduces the task of showing that $\mathcal{IBI}$, $\mathcal{IBS}$ are secure to showing that $\mathcal{SI}$ is secure in these cases.

We remark that the above gives rise to numerous schemes that are "new" in the sense that they were not provided explicitly in the literature. For example, Shamir [Sha84] defined an IBS scheme but no IBI scheme. (He even states that providing an IBI scheme is an open question.) Denoting Shamir's IBS scheme by $\mathcal{Sh}\text{-}\mathcal{IBS}$, we surface the cSI scheme $\mathcal{Sh}\text{-}\mathcal{SI}$ such that $\mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{Sh}\text{-}\mathcal{SI})) = \mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{Sh}\text{-}\mathcal{SI})) = \mathcal{Sh}\text{-}\mathcal{IBS}$. Consequently, we surface the IBI scheme $\mathcal{Sh}\text{-}\mathcal{IBI} = \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{Sh}\text{-}\mathcal{SI})$ that is related in a natural way to $\mathcal{Sh}\text{-}\mathcal{IBS}$, namely by the fact that $\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{Sh}\text{-}\mathcal{IBI}) = \mathcal{Sh}\text{-}\mathcal{IBS}$. In an analogous way we surface IBI schemes $\mathcal{Hs}\text{-}\mathcal{IBI}$ and $\mathcal{ChCh}\text{-}\mathcal{IBI}$ underlying the $\mathcal{Hs}\text{-}\mathcal{IBS}$ [Hes03] and $\mathcal{ChCh}\text{-}\mathcal{IBS}$

[CC03, Yi03] schemes, respectively.

Beside explaining existing IBI or IBS schemes, we are able to derive some new ones. We found papers in the literature [OO90, OS90, FF02] not defining IBI or IBS schemes, but proposing SI schemes that we show to be convertible. Our transforms then yield new IBI and IBS schemes that we analyze.

We feel that this systematic surfacing of implicit schemes helps to homogenize, unify, and simplify the area. Figure 3.1 summarizes the perspective that emerges. We view schemes as occurring in families. Each family has a family name $\mathcal{Name}$. At the core of the family is a cSI scheme $\mathcal{Name}\text{-}\mathcal{SI}$. The other schemes are related to it via $\mathcal{Name}\text{-}\mathcal{IBI} = \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{Name}\text{-}\mathcal{SI})$, $\mathcal{Name}\text{-}\mathcal{SS} = \mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{Name}\text{-}\mathcal{SI})$, and $\mathcal{Name}\text{-}\mathcal{IBS} = \mathsf{cSS\text{-}2\text{-}IBS}(\mathcal{Name}\text{-}\mathcal{SS})$. If $\mathcal{Name}\text{-}\mathcal{SI}$ is secure, then so are all other schemes in the family.

Results for specific schemes. In order to complete the task of obtaining security proofs for the existing and new IBI and IBS schemes we have discussed, it remains to analyze the cSI schemes underlying the families in question. This turned out to be a large task, for although in a few cases the cSI scheme was already analyzed in the literature, we found (perhaps surprisingly) that in many cases it wasn't. Additionally, we need to directly analyze two IBI schemes not based on cSI schemes, namely the DL-based scheme by Okamoto [Oka93], and a somewhat more efficient Schnorr-based [Sch90] variant that we introduce.

A summary of our results is given in Figure 3.2. Sections 3.5 and 3.6 provide scheme descriptions, more precise result statements and security proofs. Note that all proofs for SS, IBI, and IBS schemes are in the random oracle model. We highlight some of the important elements of our results here.

Section 3.5 begins by surfacing SI schemes underlying the first 13 (i.e. all but the last two) families of Figure 3.2 and shows that they are convertible, so that the picture of Figure 3.1 holds in all these cases and we need only consider security of the cSI schemes. The analysis of these schemes follows.

Easy cases are $\mathcal{FS}$, $\mathcal{ItR}$ (the iterated-root family, also known as 2^t-th root in the literature), $\mathcal{FF}$, $\mathcal{GQ}$, and $\mathcal{OKRSA}$ (an RSA-based family by Okamoto [Oka93]) where the SI schemes are already present and analyzed in previous work [FFS88, Sch96, FF02, BP02, Oka93].

The $\mathcal{Sh}\text{-}\mathcal{SI}$ scheme turns out to be a mirror-image of $\mathcal{GQ}\text{-}\mathcal{SI}$, and is interesting technically because we show that it is honest-verifier zero-knowledge (HVZK) even though it might not at first appear to be so. Based on this, we prove that it is imp-pa secure, but simple attacks show that imp-aa and imp-ca security do not hold. A slight modification $\mathcal{Sh}^*\text{-}\mathcal{SI}$ of this scheme however is not only imp-pa but also proven imp-aa and imp-ca secure under the one-more RSA assumption, so that its security is similar to that of $\mathcal{GQ}\text{-}\mathcal{SI}$ [BP02].

An attack and a fix for Girault's IBI scheme [Gir90] were proposed [SSN98], but we find attacks on the fixed scheme as well, breaking all schemes in the family.

Name	Origin	*Name-SI*			*Name-IBI*			*Name-SS*	*Name-IBS*
		imp-pa	imp-aa	imp-ca	imp-pa	imp-aa	imp-ca	uf-cma	uf-cma
$\mathcal{FS}$	IBI,IBS [FS86, FFS88]	[FS86]	[FFS88]	**I**	**I**	**I**	**I**	[PS00]	[DKXY03]
$\mathcal{ItR}$	SI, SS [OO90, OS90]	[Sch96]	[Sch96]	**U**	**I**	**I**	**U**	[PS00]	[DKXY03]
$\mathcal{FF}$	SI,SS [FF02]	[FF02]	[FF02]	[FF02]	**I**	**I**	**I**	[FF02]	[DKXY03]
$\mathcal{GQ}$	IBI, IBS [GQ89]	[GQ89]	[BP02]	[BP02]	**I**	**I**	**I**	[PS00]	[DKXY03]
$\mathcal{Sh}$	IBS [Sha84]	**P**	**A**	**A**	**I**	**A**	**A**	**I**	**I**
$\mathcal{Sh}^*$	SI	**P**	**P**	**P**	**I**	**I**	**I**	**I**	**I**
$\mathcal{OKRSA}$	SI, IBI, SS [Oka93]	[Oka93]	[Oka93]	**I**	**I**	**I**	**I**	[PS00]	[DKXY03]
$\mathcal{Gir}$	SI, IBI [Gir90, SSN98]	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
$\mathcal{SOK}$	IBS [SOK00]	**P**	**A**	**A**	**I**	**A**	**A**	**I**	**I**
$\mathcal{Hs}$	IBS [Hes03]	**P**	**P**	**P**	**I**	**I**	**I**	[Hes03]	[DKXY03]
$\mathcal{ChCh}$	IBS [CC03, Yi03]	**P**	**P**	**P**	**I**	**I**	**I**	[CC03]	[CC03]
$\mathcal{Beth}^1$	IBI [Bet88]	**P**	**U**	**U**	**I**	**U**	**U**	**I**	**I**
$\mathcal{Beth}^t$	IBI [Bet88]	**U**	**U**	**U**	**U**	**U**	**U**	**U**	**U**
$\mathcal{OkDL}$	IBI [Oka93]	**I**	**I**	**I**	**P**	**P**	**P**	**I**	**I**
$\mathcal{XDL}$	SI, IBI	**I**	**I**	**I**	**P**	**P**	**P**	**I**	**I**

Figure 3.2: Summary of security results. Column 1 is the family name of a family of schemes. Column 2 indicates which of the four member-schemes of the family existed in the literature. (The others we surface.) In the security columns, a known result is indicated via a reference to the paper establishing it. The marks **I**, **P**, and **A** all indicate new results obtained in this work. An **I** indicates a proof of security obtained by implication. (If under *Name-IBI* it means we obtain it via Theorem 3.8, if under *Name-IBS* it means we obtain it either via Corollary 3.11 or via our modified fs-I-2-S transform, if elsewhere it means it follows easily from, or is an easy extension of, existing work.) A **P** indicates a new security proof, such as a from-scratch analysis of some SI or IBI scheme. An **A** indicates an attack that we have found. A **U** indicates that the security status is unknown. In all but the last two rows, the SI scheme is convertible. The first set of schemes are factoring based, the next RSA based, the next pairing based, and the last DL based.

We prove imp-pa security of the pairing-based $\mathcal{SOK}$-$\mathcal{SI}$, $\mathcal{Hs}$-$\mathcal{SI}$ and $\mathcal{ChCh}$-$\mathcal{SI}$ schemes under the CDH assumption, and imp-aa and imp-ca security under the one-more CDH assumption. We remark that the $\mathcal{SOK}$-$\mathcal{IBS}$ scheme defined via our transforms is not that of Sakai et al. [SOK00], but is a slightly variant of it. This suggests the value of our framework, for it is unclear whether the IBS scheme of Sakai et al. can be proven uf-cma secure, whereas Corollary 3.11 implies that our variant $\mathcal{SOK}$-$\mathcal{IBS}$ is uf-cma secure.

Since the discrete logarithm function has no known trapdoor it is not an obvious starting point for IBI schemes, but some do exist. Beth's (unproven) IBI scheme $\mathcal{Beth}^t$-$\mathcal{IBI}$ [Bet88] is parameterized with a "key multiplicity" $t \in \{1, 2, \ldots\}$ and is based on ElGamal signatures. The proof of convertibility of the $\mathcal{Beth}^1$-$\mathcal{SI}$ scheme we surface is interesting in that it exploits the existential forgeability of ElGamal signatures [El 84]. We prove that $\mathcal{Beth}^1$-$\mathcal{SI}$ is imp-pa secure if the hashed-message ElGamal signature scheme is universally unforgeable under no-message attack in the random oracle model. We were unable to either prove secure or break $\mathcal{Beth}^1$-$\mathcal{SI}$ under active and concurrent attacks. We were also unable to prove any security results at all about the $\mathcal{Beth}^t$ family for $t > 1$.

Exceptions. The last two rows of Figure 3.2 represent cases where our framework does not apply and direct analyses are needed. The first such case is an unproven discrete logarithm based IBI scheme $\mathcal{OKDL}$-$\mathcal{IBI}$ due to Okamoto [Oka93], which introduces an interesting SS-based method for constructing IBI schemes and instantiates it with a discrete logarithm-based SS scheme presented in the same paper. We were unable to surface any cSI scheme that under cSI-2-IBI maps to $\mathcal{OKDL}$-$\mathcal{IBI}$. ($\mathcal{OKDL}$-$\mathcal{IBI}$ can be "dropped" in a natural way to a SI scheme $\mathcal{OKDL}$-$\mathcal{SI}$, but the latter does not appear to be convertible.) However, we show that $\mathcal{OKDL}$-$\mathcal{IBI}$ is nevertheless imp-pa, imp-aa and imp-ca secure assuming the hardness of the discrete logarithm problem. This direct proof is probably the most technical in this chapter and uses the security of Okamoto's well-known discrete logarithm based SS scheme under a weakened notion of non-malleability [SPMLS02], which is established via an extension of the result of Abdalla et al. [AABN02] combined with results from Okamoto [Oka93]. We also present a new IBI scheme $\mathcal{XDL}$-$\mathcal{IBI}$ that is based on the technique underlying $\mathcal{OKDL}$-$\mathcal{IBI}$ but uses Schnorr signatures [Sch90] instead of Okamoto signatures. It is slightly more efficient than $\mathcal{OKDL}$-$\mathcal{IBI}$. Security results are analogous to those above.

Since they do not originate from cSI schemes, the IBS schemes obtained as fs-I-2-S($\mathcal{OKDL}$-$\mathcal{IBI}$) and fs-I-2-S($\mathcal{XDL}$-$\mathcal{IBI}$) cannot be proven secure based merely on the security properties of the IBI schemes. However, we can apply our modified efs-IBI-2-IBS transform to $\mathcal{OKDL}$-$\mathcal{IBI}$ and $\mathcal{XDL}$-$\mathcal{IBI}$ to obtain uf-cma IBS schemes.

3.2 Security Notions

3.2.1 Identification Schemes

Standard identification schemes. A *standard identification (SI) scheme* is a tuple $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ of algorithms where Kg is the randomized polynomial-time key generation algorithm, and P and V are polynomial-time interactive algorithms called the prover and verifier algorithms, respectively. In an initialization step, the prover runs $\mathsf{Kg}(1^k)$, where k is a security parameter, to obtain a key pair (pk, sk), and publishes the public key pk while keeping the secret key sk private. In the interactive identification protocol, the prover runs P with initial state sk, and the verifier runs V with initial state pk. The first and last messages of the protocol belong to the prover. The protocol ends when V enters either the `acc` or `rej` state. We require that for all $k \in \mathbb{N}$ and for all $(pk, sk) \in [\mathsf{Kg}(1^k)]$, the result of the interaction between P (initialized with sk) and V (initialized with pk) is `acc` with overwhelming probability. This correctness requirement is necessary to exclude trivial schemes such as the scheme that always rejects.

Security of SI schemes. We first give an intuitive description of the game defining security of SI schemes, and then proceed to a more formal description. The security experiment runs in two stages. The first is the *learning* stage, in which the adversary is either allowed to listen in on conversations between an honest prover and an honest verifier (passive attack [FFS88]), or is allowed to play the role of a *cheating verifier* in the interaction with honest provers (active [FFS88] and concurrent [BP02] attack). In an active attack, the adversary can only interact with a single prover at the same time, while a concurrent attack allows the adversary to arbitrarily interleave the conversations with any number of prover instances it chooses. When it decides it gathered enough information, the adversary announces it is ready to enter the second stage of the game: the *impersonation* stage. In this stage, the adversary is deprived from its previous powers and is faced with an honest verifier instead. It now has to play the role of a *cheating prover* and try to make the honest verifier accept using the knowledge it acquired during the previous stage of the game. The adversary is said to win the game if it succeeds in doing so.

Note that an identification scheme is not meant to protect against a "man-in-the-middle" attack, where the adversary makes a verifier accept by forwarding messages from an honest prover it is simultaneously interacting with. An identification scheme provides entity authentication, which merely guarantees that the entity is "alive" at the moment of verification. No message is being authenticated during the protocol as in signature schemes, and the participants do not share a secret key at the end of the protocol as in authenticated key-establishment protocols.

More formally, an SI adversary A is modelled as a pair of algorithms $(\mathsf{CV}, \mathsf{CP})$

called the *cheating verifier* and the *cheating prover*. The experiment first chooses keys (pk, sk) via $\mathsf{Kg}(1^k)$ and then runs CV on input pk. For a passive attack (pa), CV has access to a conversation oracle CONV, a query to which returns the transcript of a fresh conversation between P and V:

$St_{\mathsf{P}} \leftarrow sk$; $R_{\mathsf{P}} \xleftarrow{R} \{0,1\}^{\rho_{\mathsf{P}}}$
$St_{\mathsf{V}} \leftarrow (pk, R_{\mathsf{V}})$; $R_{\mathsf{V}} \xleftarrow{R} \{0,1\}^{\rho_{\mathsf{V}}}$
$T \leftarrow \varepsilon$; $M_{\rm in} \leftarrow \varepsilon$
Repeat
  $(M_{\rm out}, St_{\mathsf{P}}) \leftarrow \mathsf{P}(M_{\rm in}, St_{\mathsf{P}}, R_{\mathsf{P}})$
  $(M_{\rm in}, St_{\mathsf{V}}) \leftarrow \mathsf{V}(M_{\rm out}, St_{\mathsf{V}}, R_{\mathsf{V}})$
  $T \leftarrow T \| M_{\rm in} \| M_{\rm out}$
Until $St_{\mathsf{V}} \in \{\mathtt{acc}, \mathtt{rej}\}$
Return T.

For an active attack (aa) or concurrent attack (ca), CV gets a prover oracle PROV. Upon a query (M, s) where M is a message and s is a session number, the PROV oracle runs the prover algorithm P using M as an incoming message and returns the prover's outgoing message while maintaining the prover's state associated with the session s across invocations. (For each new session, PROV uses fresh random coins to start the prover, initializing it with sk.) The difference between active and concurrent attacks is that the former allows only a single prover to be active at the same time. Eventually, CV halts with some output that is given as initial state to interactive algorithm CP, and A wins if the interaction between CP and V (the latter initialized with pk) leads the latter to accept. For $\mathrm{atk} \in \{\mathrm{pa}, \mathrm{aa}, \mathrm{ca}\}$, the imp-atk advantage of A in attacking $\mathcal{SI}$ is written as $\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{SI},\mathsf{A}}(k)$ and is defined to be the probability of A winning in the above experiment. We say that $\mathcal{SI}$ is an atk-*secure SI scheme* if $\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{SI},\mathsf{A}}(\cdot)$ is a negligible function for every A whose two component algorithms have running time polynomial in k.

IDENTITY-BASED IDENTIFICATION SCHEMES. An *identity-based identification (IBI) scheme* is a four-tuple $\mathit{IBI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$ of polynomial-time algorithms. The trusted, key-issuing authority runs the *master-key generation* algorithm MKg on input 1^k, where k is a security parameter, to obtain a master public and secret key pair (mpk, msk). It can then run the *user-key generation* algorithm UKg on inputs msk and the identity $I \in \{0,1\}^*$ of a user to generate for this user a secret user key usk which is then assumed to be securely communicated to the user. In the interactive identification protocol, the prover with identity I runs the interactive algorithm $\overline{\mathsf{P}}$ with initial state usk, and the verifier runs $\overline{\mathsf{V}}$ with initial state mpk, I. The first and last messages of the protocol belong to the prover. The protocol ends when $\overline{\mathsf{V}}$ enters either the $\mathtt{acc}$ or the $\mathtt{rej}$ state. In the random oracle model, $\mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}}$ additionally have oracle access to a random function H whose range may depend on mpk. Correctness requires

Oracle $\textsc{Init}(I)$
If $I \in CU \cup HU$ then return $\bot$
$usk[I] \xleftarrow{R} \mathsf{UKg}(msk, I)$
$HU \leftarrow HU \cup \{I\}$
Return 1

Oracle $\textsc{Corr}(I)$
If $I \notin HU$ then return $\bot$
$CU \leftarrow CU \cup \{I\}$
$HU \leftarrow HU \setminus \{I\}$
Return $usk[I]$

Oracle $\textsc{Conv}(I)$
If $I \notin HU$ then return $\bot$
Pick random coins $\rho_{\overline{\mathsf{P}}}$ for $\overline{\mathsf{P}}$
Pick random coins $\rho_{\overline{\mathsf{V}}}$ for $\overline{\mathsf{V}}$
$St_{\overline{\mathsf{P}}} \leftarrow (usk[I], \rho_{\overline{\mathsf{P}}})$
$St_{\overline{\mathsf{V}}} \leftarrow (mpk, I, \rho_{\overline{\mathsf{V}}})$
$M_{\text{in}} \leftarrow \varepsilon$; $C \leftarrow \varepsilon$
While $(St_{\overline{\mathsf{V}}} \notin \{\mathtt{acc}, \mathtt{rej}\})$ do
$(M_{\text{out}}, St_{\overline{\mathsf{P}}}) \leftarrow \overline{\mathsf{P}}(M_{\text{in}}, St_{\overline{\mathsf{P}}})$
$(M_{\text{in}}, St_{\overline{\mathsf{V}}}) \leftarrow \overline{\mathsf{V}}(M_{\text{out}}, St_{\overline{\mathsf{V}}})$
$C \leftarrow C \| M_{\text{out}} \| M_{\text{in}}$
Return C

Oracle $\textsc{Prov}(I, s, M_{\text{in}})$
If $I \notin HU$ then return $\bot$
If $(I, s) \notin PID$ then
If atk = aa then
$PID \leftarrow \{(I, s)\}$
If atk = ca then
$PID \leftarrow PID \cup \{(I, s)\}$
Pick random coins ρ for $\overline{\mathsf{P}}$
$St_{\overline{\mathsf{P}}}[I, s] \leftarrow (usk[I], \rho)$
$(M_{\text{out}}, St_{\overline{\mathsf{P}}}[I, s]) \leftarrow \overline{\mathsf{P}}(M_{\text{in}}, St_{\overline{\mathsf{P}}}[I, s])$
Return M_{out}

Experiment $\mathbf{Exp}^{\text{imp-atk}}_{I\mathcal{BI},\overline{\mathsf{A}}}(k)$ // atk $\in \{\text{pa}, \text{aa}, \text{ca}\}$
Parse A as $(\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$; $(mpk, msk) \xleftarrow{R} \mathsf{MKg}(1^k)$
$HU \leftarrow \emptyset$; $CU \leftarrow \emptyset$; $PID \leftarrow \emptyset$ // *set of honest users, set of corrupted users and set of running prover sessions*
If atk = pa then $(J, St_{\overline{\mathsf{CP}}}) \leftarrow \overline{\mathsf{CV}}(mpk : \textsc{Init}(\cdot), \textsc{Corr}(\cdot), \textsc{Conv}(\cdot))$
Else $(J, St_{\overline{\mathsf{CP}}}) \leftarrow \overline{\mathsf{CV}}(mpk : \textsc{Init}(\cdot), \textsc{Corr}(\cdot), \textsc{Prov}(\cdot, \cdot, \cdot))$
If $J \notin HU$ then return 0
$HU \leftarrow HU \setminus \{J\}$; $CU \leftarrow CU \cup \{J\}$; $St_{\overline{\mathsf{V}}} \leftarrow (mpk, J)$; $M_{\text{in}} \leftarrow \varepsilon$
Repeat
If atk = pa then
$(M_{\text{out}}, St_{\overline{\mathsf{CP}}}) \leftarrow \overline{\mathsf{CP}}(M_{\text{in}}, St_{\overline{\mathsf{CP}}} : \textsc{Init}(\cdot), \textsc{Corr}(\cdot), \textsc{Conv}(\cdot))$
Else
$(M_{\text{out}}, St_{\overline{\mathsf{CP}}}) \leftarrow \overline{\mathsf{CP}}(M_{\text{in}}, St_{\overline{\mathsf{CP}}} : \textsc{Init}(\cdot), \textsc{Corr}(\cdot), \textsc{Prov}(\cdot, \cdot, \cdot))$
$(M_{\text{in}}, St_{\overline{\mathsf{V}}}) \leftarrow \overline{\mathsf{V}}(M_{\text{out}}, St_{\overline{\mathsf{V}}})$
Until $St_{\overline{\mathsf{V}}} \in \{\mathtt{acc}, \mathtt{rej}\}$
If $St_{\overline{\mathsf{V}}} = \mathtt{acc}$ then return 1 else return 0

Figure 3.3: Oracles given to adversary attacking IBI scheme $I\mathcal{BI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$, and experiment used to define imp-atk security of the scheme.

that for all $k \in \mathbb{N}$, $I \in \{0,1\}^*$, $(mpk, msk) \in [\mathsf{MKg}(1^k)]$, for all functions H with appropriate domain and range, and for all $usk \in [\mathsf{UKg}(msk, I : \text{H})]$, the result of

the interaction between $\overline{\mathsf{P}}$ (initialized with usk) and $\overline{\mathsf{V}}$ (initialized with mpk, I) is that $\overline{\mathsf{V}}$ accepts with probability one.

Security of IBI schemes. We first provide the formal definitions and then the explanations. Let $k \in \mathbb{N}$ be a security parameter, $\mathcal{IBI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$ be an IBI scheme, and $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ be an adversary. Consider the experiment of Figure 3.3. The type of attack $\mathrm{atk} \in \{\mathrm{pa}, \mathrm{aa}, \mathrm{ca}\}$ is a parameter, and the adversary has access to the oracles shown in the same figure. Let $\mathrm{atk} \in \{\mathrm{pa}, \mathrm{aa}, \mathrm{ca}\}$. The imp-atk *advantage* of $\overline{\mathsf{A}}$ in attacking $\mathcal{IBI}$ is

$$\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) \;=\; \Pr\left[\,\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) = 1\,\right].$$

We say that $\mathcal{IBI}$ is an imp-atk-*secure IBI scheme* if $\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(\cdot)$ is negligible for every polynomial-time $\overline{\mathsf{A}}$.

The main difference with the SI experiment is that $\overline{\mathsf{A}}$ can initialize or corrupt identities of its choice through the Init and Corr oracles. This models the possibility that the adversary itself is a user of the system, possibly even colluding with other users. HU is the set of honest users, and CU is the set of corrupted users. At the end of its execution, $\overline{\mathsf{CV}}$ transfers its state to $\overline{\mathsf{CP}}$ and outputs an uncorrupted identity J. In the second stage, $\overline{\mathsf{CP}}$ will try to impersonate J. An element of this definition worth drawing attention to is that we have allowed $\overline{\mathsf{CP}}$ to query the same oracles as $\overline{\mathsf{CV}}$. This allows $\overline{\mathsf{CP}}$ to initialize, corrupt, interact with, or see conversations involving certain identities depending on the challenge it gets from the verifier. The only restriction is that $\overline{\mathsf{CP}}$ *cannot* submit queries involving J because otherwise impersonating J would become trivial. The restrictions are all enforced by the oracles themselves. (At the end of the first stage, J is removed from HU and added to CU.)

3.2.2 Signature Schemes

We recall security definitions for SS and IBS schemes [GMR88, CC03, DKXY03].

Standard signature schemes and their security. A *standard signature (SS) scheme* $\mathcal{SS}$ is a triple of algorithms $(\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$. On input 1^k, where k is the security parameter, the randomized key generation algorithm Kg returns a fresh key pair (pk, sk). On input a secret key sk and a message M, the possibly randomized signing algorithm Sign returns a signature σ. On input pk, M, and a signature σ, the deterministic verification algorithm Vf returns 1 to indicate that σ is a valid signature for M under pk, or returns 0 otherwise. In the random oracle model, the last two algorithms have access to a random oracle H. We require that, for all $(pk, sk) \in \mathsf{Kg}(1^k)$ and for all messages $M \in \{0,1\}^*$, it is always the case that $\mathsf{Vf}(pk, M, \mathsf{Sign}(sk, M)) = 1$.

We use the security notion of existential unforgeability under chosen-message attack (uf-cma) [GMR88], which is the following. The experiment generates a

fresh key pair $(pk, sk) \xleftarrow{R} \mathsf{Kg}(1^k)$ and runs the adversary F, also called the *forger*, on input pk. The forger has access to a signing oracle SIGN that on input message M returns a signature $\sigma \xleftarrow{R} \mathsf{Sign}(sk, M)$. At the end of its execution, F outputs a forgery consisting of a message M and a forged signature σ. The forger wins the game if $\mathsf{Vf}(pk, M, \sigma) = 1$ while M was not previously queried to the SIGN oracle. We define the uf-cma advantage of F in breaking $\mathcal{SS}$, denoted as $\mathbf{Adv}^{\text{uf-cma}}_{\mathsf{F},\mathcal{SS}}(k)$, to be the probability that F wins the game.

While existential unforgeability under chosen-message attack is commonly regarded as the most useful security notion for SS schemes, we will occasionally use other notions in proofs or attacks. Variations exist on both the adversarial goal and the attack model. As for the latter, we mention the less powerful (resulting in weaker security notions) *known-message attack*, where the adversary sees signatures of messages chosen by the experiment, and *no-message attack*, where the adversary doesn't get to see any signatures at all. Alternative adversarial goals include the weaker notion of *universal unforgeability* where the forger has to be able to sign *any* message instead of just *some* message, and the stronger notion of *non-malleability* [SPMLS02] that also accepts a new (different) signature on a previously signed message as a valid forgery.

IDENTITY-BASED SIGNATURE SCHEMES AND THEIR SECURITY. An *identity-based signature (IBS) scheme* is a tuple $\mathcal{IBS} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{Sign}}, \overline{\mathsf{Vf}})$ of polynomial-time randomized algorithms. The trusted, key-issuing authority runs the master-key generation algorithm MKg on input 1^k, where k is a security parameter, to obtain a master public and secret key pair (mpk, msk). It can then run the user-key generation algorithm UKg on msk and the identity $I \in \{0,1\}^*$, thus generating for the user I a secret key usk which is then securely communicated to I. On input usk and a message M, the signing algorithm $\overline{\mathsf{Sign}}$ returns a signature of M. On input mpk, I, M, and a signature σ, the verification algorithm $\overline{\mathsf{Vf}}$ returns 1 to indicate that σ is a valid signature for M under mpk for identity I, or returns 0 otherwise. We require that, for all $k \in \mathbb{N}$, $M \in \{0,1\}^*$, and $I \in \{0,1\}^*$,

$$\Pr\left[\begin{array}{l} \overline{\mathsf{Vf}}(mpk, I, M, \sigma) = 1 \mid (mpk, msk) \xleftarrow{R} \mathsf{MKg}(1^k)\ ; \\ usk \xleftarrow{R} \mathsf{UKg}(msk, I)\ ;\ \sigma \xleftarrow{R} \overline{\mathsf{Sign}}(usk, M) \end{array}\right] = 1\ .$$

We first provide the formal definition and then the explanations. Let $k \in \mathbb{N}$ be a security parameter, $\mathcal{IBS} = (\mathsf{MKg}, \mathsf{UKg}, \mathsf{Sign}, \mathsf{Vf})$ be an IBS scheme, and $\overline{\mathsf{F}}$ be an adversary. Consider the experiment below. The adversary has access to the oracles shown in Figure 3.4:

Oracle $\textsc{Init}(I)$	Oracle $\textsc{Sign}(I, M)$
If $I \in CU \cup HU$ then return $\perp$	If $I \notin HU$ then return $\perp$
$usk[I] \xleftarrow{R} \mathsf{UKg}(msk, I)$	$\sigma \xleftarrow{R} \overline{\mathsf{Sign}}(usk[I], M)$
$MSG[I] \leftarrow \emptyset$; $HU \leftarrow HU \cup \{I\}$	$MSG[I] \leftarrow MSG[I] \cup \{M\}$
Return 1	Return σ

Figure 3.4: Oracles provided to an adversary attacking an identity-based signature scheme $\mathcal{IBS} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{Sign}}, \overline{\mathsf{Vf}})$. The oracle $\textsc{Corr}$ is the same as that in Figure 3.3 and thus is not shown here.

Experiment $\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{IBS},\overline{\mathsf{F}}}(k)$
$(mpk, msk) \xleftarrow{R} \mathsf{MKg}(1^k)$
$HU \leftarrow \emptyset$; $CU \leftarrow \emptyset$
$(I, M, \sigma) \xleftarrow{R} \overline{\mathsf{F}}(mpk : \textsc{Init}(\cdot), \textsc{Sign}(\cdot,\cdot), \textsc{Corr}(\cdot))$
If ($I \in HU$ and $\mathsf{Vf}(mpk, I, M, \sigma) = 1$ and $M \notin MSG[I]$)
then return 1 else return 0

The uf-cma advantage of $\overline{\mathsf{F}}$ in attacking $\mathcal{IBS}$ is

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{IBS},\overline{\mathsf{F}}}(k) \;=\; \Pr\left[\,\mathbf{Exp}^{\text{uf-cma}}_{\mathcal{IBS},\overline{\mathsf{F}}}(k) = 1\,\right].$$

We say that $\mathcal{IBS}$ is a *secure IBS scheme* if $\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{IBS},\overline{\mathsf{F}}}(\cdot)$ is negligible for every polynomial-time adversary $\overline{\mathsf{F}}$.

Via $\textsc{Init}(I)$, the adversary $\overline{\mathsf{F}}$ can create a user I. Invisibly to the adversary, a secret key denoted $usk[I]$ is assigned to I. Via $\textsc{Sign}(I, M)$, it can obtain I's signature on a message M of its choice. Via $\textsc{Corr}(I)$, it can compromise I's secret key $usk[I]$. To win, $\overline{\mathsf{F}}$ must output the identity I of an uncorrupted user, a message M, and a signature σ such that $\overline{\mathsf{Vf}}(mpk, I, M, \sigma) = 1$ while I did not previously sign M. Here, HU is the set of honest users, CU is the set of corrupted users, and $MSG[I]$ is the set of messages that I has signed. As always, the uf-cma advantage of $\overline{\mathsf{F}}$ is the probability that it wins the game.

3.3 Certification-Based IBI and IBS

There is a natural way to construct IBI and IBS schemes using certificates. The idea is simply that the authority can issue a certificate, consisting of a signature of a user's identity and "public key," the latter being the value the authority chooses and provides to the user along with a matching secret key. By including this public key and certificate in a signature under the authority's secret key, verification of this signature becomes possible given only the authority public key and identity of the user, and hence is identity-based.

We believe that the scheme is folklore, but it is worth detailing and proving for several reasons. One is that it shows that IBI and IBS are achievable without random oracles (by instantiating the underlying SS-scheme with a scheme that is secure in the standard model [GHR99, CS00]). All the practical schemes we consider do use random oracles. Another is that this scheme is a benchmark relative to which practical schemes should measure their efficiency.

Note that no such simple trick works for identity-based encryption without disrupting the communication pattern that the sender initiates the communication by sending a ciphertext encrypted with the recipient's public key. A certificate-based IBE scheme can only be constructed at the price of an extra round of interaction.

We now provide some details, showing the design of an IBI scheme based on any SI scheme and any SS scheme. One can analogously design an IBS scheme based on any SS scheme. Let $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ be a SI scheme, and let $\mathcal{SS} = (\mathsf{SKg}, \mathsf{Sign}, \mathsf{Vf})$ be a SS scheme. We associate to them an IBI scheme $\mathcal{IBI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$ whose constituent algorithms are as follows. The master key generation algorithm MKg is simply SKg, so that the master secret key msk can be used to produce signatures verifiable under mpk. To issue a secret key usk to a user with identity I, the authority first runs $\mathsf{Kg}(1^k)$ to obtain a public and secret key pair (pk, sk) for the SI scheme. It then creates the certificate $cert \leftarrow (pk, \mathsf{Sign}(msk, pk \| I))$. (The identity need not be included in the certificate, as it is passed as a parameter to the verification algorithm anyway.) It sets $usk \leftarrow (sk, cert)$ and sends the latter to I. The interactive algorithm $\overline{\mathsf{P}}$, run by I to identify itself, runs P, initializing the latter with sk, and includes $cert$ in the first flow sent to the verifier. The interactive algorithm $\overline{\mathsf{V}}$, run by the verifier, has inputs mpk, I. In the first move it receives $cert$ along with any information that P has sent on its first move. It then verifies the signature on the certificate $cert$. If the certificate is invalid, $\overline{\mathsf{V}}$ halts and rejects. Otherwise, it runs V, initializing the latter with pk. It accepts if V accepts. Construction 3.1 and Figure 3.5 describe this construction in detail. Theorem 3.2 says that the construction is secure.

Construction 3.1 (Certificate-based IBI) Given a standard identification scheme $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ and a (standard) signature scheme $\mathcal{SS} = (\mathsf{SKg}, \mathsf{Sign}, \mathsf{Vf})$, we associate to them an IBI scheme $\mathcal{IBI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$ whose constituent algorithms are depicted in Figure 3.5. ∎

Theorem 3.2 (Security of Certificate-based IBI) *Let $\mathcal{SI}$ be a SI scheme, and $\mathcal{SS}$ a* uf-cma *secure SS scheme. Let $\mathcal{IBI}$ be the corresponding certificate-based IBI scheme as per Construction 3.1. If $\mathcal{SI}$ is* imp-atk *secure then $\mathcal{IBI}$ is* imp-atk *secure, for any* atk $\in \{\mathrm{pa}, \mathrm{aa}, \mathrm{ca}\}$. ∎

$\mathsf{MKg}(1^k)$:
$(mpk, msk) \leftarrow \mathsf{SKg}(1^k)$
Return (mpk, msk)

$\mathsf{UKg}(msk, I; k)$:
$(pk, sk) \leftarrow \mathsf{Kg}(1^k)$
$cert \leftarrow (pk, \mathsf{Sign}(msk, pk \| I))$
$usk \leftarrow (sk, cert)$
Return usk

$\overline{\mathsf{P}}(M_{\text{in}}, St_{\overline{\mathsf{P}}})$:
If $St_{\overline{\mathsf{P}}}$ parses as $(sk, cert)$ then
$(M_{\text{out}}, St_{\overline{\mathsf{P}}}) \leftarrow \mathsf{P}(M_{\text{in}}, sk)$
$M_{\text{out}} \leftarrow cert \| M_{\text{out}}$
Else $(M_{\text{out}}, St_{\overline{\mathsf{P}}}) \leftarrow \mathsf{P}(M_{\text{in}}, St_{\overline{\mathsf{P}}})$
Return $(M_{\text{out}}, St_{\overline{\mathsf{P}}})$

$\overline{\mathsf{V}}(M_{\text{in}}, St_{\overline{\mathsf{V}}})$:
If M_{in} parses as $cert \| M$ then
Parse $cert$ as (pk, σ)
Parse $St_{\overline{\mathsf{V}}}$ as (mpk, I)
$M_{\text{in}} \leftarrow M$
If $\mathsf{Vf}(mpk, pk \| I, \sigma) \neq 1$ then
$St_{\overline{\mathsf{V}}} \leftarrow \texttt{rej}$ else $St_{\overline{\mathsf{V}}} \leftarrow pk$
$(M_{\text{out}}, St_{\overline{\mathsf{V}}}) \leftarrow \mathsf{V}(M_{\text{in}}, St_{\overline{\mathsf{V}}})$

Figure 3.5: A certificate-based IBI scheme $\mathcal{IBI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$ constructed from a standard identification scheme $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ and a digital signature scheme $\mathcal{SS} = (\mathsf{SKg}, \mathsf{Sign}, \mathsf{Vf})$.

Proof: Given $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ attacking $\mathcal{IBI}$, we construct a forger algorithm F attacking $\mathcal{SS}$ and an impersonator A attacking $\mathcal{SI}$. Impersonator $\overline{\mathsf{A}}$ can impersonate a user in two ways: either by forging a certificate, or by reusing an existing certificate but exploiting a weakness in the underlying SI scheme. The first type of impersonation can be used to forge signatures for $\mathcal{SS}$, the second type can be used to break the $\mathcal{SI}$ scheme.
We first present the forger algorithm F. It gets a public key mpk for $\mathcal{SS}$ as input, and access to a SIGN oracle initialized with the corresponding secret key. F runs $\overline{\mathsf{CV}}$ on input mpk, answering its oracle queries using the algorithms of $\mathcal{IBI}$ exactly as the real experiment would, but calling its $\text{SIGN}(pk_I \| I)$ to generate certificates associating identities I to their public key pk_I. It is clear that this is a perfect simulation of $\overline{\mathsf{CV}}$'s environment. When $\overline{\mathsf{CV}}$ halts and announces the identity J that will be attacked, F runs $\overline{\mathsf{CP}}$ on the initial state passed to it by $\overline{\mathsf{CV}}$. The forger F answers $\overline{\mathsf{CP}}$'s oracle queries as before until the cheating prover outputs its first message $cert \| M$. The certificate is parsed as (pk, σ). In the event $\mathbf{E}$ that $pk \neq pk_J$ (where pk_J is the public key that F once generated for identity J), F outputs σ as its forgery for message $pk \| J$, otherwise F aborts. We can bound the advantage of F in breaking $\mathcal{SS}$ from below as

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{F}}(k) \geq \Pr\left[\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) = 1 \wedge \mathbf{E}\right]. \tag{3.1}$$

We now describe the impersonator $\mathsf{A} = (\mathsf{CV}, \mathsf{CP})$ attacking the underlying $\mathcal{SI}$

scheme. Algorithm CV gets a public key pk as input, and has access to a conversation oracle $\textsc{Conv}$ (passive attack) or a prover oracle $\textsc{Prov}(\cdot)$ (active and concurrent attack). Its strategy is to simulate $\overline{\mathsf{CV}}$'s environment exactly as in a real attack, *except* for one identity I_{guess} that it will simulate using its own oracles and that it hopes to be the identity J that $\overline{\mathsf{CP}}$ will attack in the second stage of the game. Guessing an identity I_{guess} from the infinite set $\{0,1\}^*$ is of course infeasible. We know however that $\overline{\mathsf{CV}}$ must first initialize the identity using a query to the $\textsc{Init}$ oracle, so instead, CV chooses a random integer $q_{\text{guess}} \xleftarrow{R} \{1, \ldots, \mathbf{Q}^{\textsc{Init}}_{\overline{\mathsf{CV}}}\}$ and assigns to I_{guess} the argument of $\overline{\mathsf{CV}}$'s q_{guess}-th $\textsc{Init}$ query. For all identities other than I_{guess}, CV simulates $\overline{\mathsf{CV}}$'s oracles exactly as the real experiment would, generating one fresh key pair (mpk, msk) for $\mathcal{SS}$ to sign certificates, and generating a fresh $\mathcal{SI}$ key pair (pk_I, sk_I) for each initialized identity I. In reply to $\overline{\mathsf{CV}}$'s q_{guess}-th initialization query, however, CV creates a certificate linking the public key pk to identity I_{guess}. To answer $\overline{\mathsf{CV}}$'s $\textsc{Conv}(I_{\text{guess}})$ or $\textsc{Prov}(I_{\text{guess}}, \cdot, \cdot)$ queries, CV queries its own $\textsc{Conv}$ or $\textsc{Prov}(\cdot, \cdot)$ oracle and forwards the reply (concatenated with the identity's certificate on the first move) to $\overline{\mathsf{CV}}$. If $\overline{\mathsf{CV}}$ asks to corrupt I_{guess}, CV gives up. It is clear that the simulation of $\overline{\mathsf{CV}}$'s environment is perfect as long as I_{guess} remains uncorrupted. From $\overline{\mathsf{CV}}$'s point of view, all initiated identities are alike, so it has no more reason to corrupt I_{guess} than to corrupt any other identity. Since the case we are interested in is when I_{guess} becomes the identity under attack in the second stage of the game, and since only uncorrupted identities can be attacked, such query will not occur whenever it actually matters to CV.

At the end of its execution, $\overline{\mathsf{CV}}$ outputs the identity to be attacked J and state information $St_{\overline{\mathsf{CP}}}$. If $J \neq I_{\text{guess}}$, CV gives up. Otherwise, it announces to be ready to proceed to the second stage of the game, outputting state information for CP containing pk, $St_{\overline{\mathsf{CP}}}$ and all keys generated to answer $\overline{\mathsf{CV}}$'s oracle queries. Algorithm CP runs $\overline{\mathsf{CP}}(\varepsilon, St_{\overline{\mathsf{CP}}})$, answering $\overline{\mathsf{CP}}$'s oracle queries in the same way CV did for $\overline{\mathsf{CV}}$. (Note that CP is not allowed to make queries involving J anymore.) Let $cert \| M$ be the first message output by $\overline{\mathsf{CV}}$. In the event $\mathbf{E}$ that the public key pk' included in the certificate is different from pk, CP gives up. Otherwise, it sends M as its own first message to the honest verifier V, and it keeps sending messages back and forth between $\overline{\mathsf{CP}}$ and V until the latter accepts or rejects. It is clear that if $\overline{\mathsf{CP}}$'s impersonation is successful, V will accept, so we can bound the advantage of A as

$$\begin{aligned}
\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{SI},\mathsf{A}}(k) &\geq \Pr\left[\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) = 1 \wedge \overline{\mathbf{E}} \wedge J = I_{\text{guess}}\right] \\
&= \Pr\left[\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) = 1 \wedge \overline{\mathbf{E}}\right] \cdot \Pr\left[J = I_{\text{guess}}\right] \\
&= \frac{1}{\mathbf{Q}^{\textsc{Init}}_{\overline{\mathsf{CV}}}} \cdot \Pr\left[\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) = 1 \wedge \overline{\mathbf{E}}\right], \qquad (3.2)
\end{aligned}$$

where the second line is true because the event $J = I_{\text{guess}}$ is independent of the other two since $\overline{\mathsf{CV}}$'s view is independent of CV's choice for I_{guess}, and the last line is true because $\overline{\mathsf{CV}}$ guesses correctly with probability $1/\mathbf{Q}^{\text{Init}}_{\overline{\mathsf{CV}}}$.
By combining Equation (3.1) and Equation (3.2), the advantage of an adversary $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ attacking $\mathcal{IBI}$ can be bounded as

$$\begin{aligned}
\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) &= \Pr\left[\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k)\right] \\
&= \Pr\left[\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) \wedge \mathbf{E}\right] + \Pr\left[\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) \wedge \overline{\mathbf{E}}\right] \\
&\leq \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{F}}(k) + \mathbf{Q}^{\text{Init}}_{\overline{\mathsf{CV}}} \cdot \mathbf{Adv}^{\text{imp-atk}}_{\mathcal{SI},\mathsf{A}}(k) \ ,
\end{aligned}$$

which proves the theorem. ∎

3.4 Transformations between Schemes

3.4.1 The Fiat-Shamir Transform

So-called *canonical* SI schemes can be transformed into signature schemes using the Fiat-Shamir transform [FS86], referred to as the fs-I-2-S transform here. A standard identification scheme $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ is said to be *canonical* if it follows a three-move structure where the prover initiates the communication with a commitment *Cmt* distributed uniformly over a set $\text{CmtSet}(sk)$ possibly depending on the secret key, the verifier sends back a challenge *Ch* chosen uniformly from a set $\text{ChSet}(pk)$ possibly depending on the public key, and the prover complies with a response *Rsp*. The verifier's decision to accept or reject is a deterministic function $Dec(pk, Cmt\|Ch\|Rsp)$ of the public key and the communication transcript. Let the *commitment length* be the greatest integer $\beta(k)$ such that $|\text{CmtSet}(sk)| \geq 2^{\beta(k)}$ for all $(pk, sk) \in [\mathsf{Kg}(1^k)]$. The scheme is said to be *non-trivial* if $\beta(k)$ is a super-logarithmic function in k. [1]

Construction 3.3 (The fs-I-2-S Transform [FS86]) Let $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ be a non-trivial canonical standard identification scheme as defined above with challenge set function ChSet and decision function *Dec*. We associate to $\mathcal{SI}$ a standard signature scheme $\mathcal{SS} = \text{fs-I-2-S}(\mathcal{SI}) = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ that has the same key generation algorithm as $\mathcal{SI}$. The signing and verification algorithms have access to a random oracle $\mathrm{H} : \{0,1\}^* \to \text{ChSet}(pk)$ and are defined as follows:

[1] The canonicity definition used here is more restrictive than the one used by Abdalla et al. [AABN02], which allows *Cmt* to be chosen according to *any* distribution over $\text{CmtSet}(sk)$. This however complicates the non-triviality condition, requiring $\beta(k)$ to be defined as the *min-entropy* of the distribution. Since all schemes treated in this thesis have uniformly distributed commitments, we restrict ourselves to the simpler definition here.

Algorithm $\mathsf{Sign}(sk, M : \mathrm{H})$	Algorithm $\mathsf{Vf}(pk, M, \sigma : \mathrm{H})$
$(Cmt, St_{\mathsf{P}}) \xleftarrow{R} \mathsf{P}(\varepsilon, sk)$	Parse σ as $Cmt \| Rsp$
$Ch \leftarrow \mathrm{H}(Cmt \| M)$	$Ch \leftarrow \mathrm{H}(Cmt \| M)$
$(Rsp, St_{\mathsf{P}}) \xleftarrow{R} \mathsf{P}(Ch, St_{\mathsf{P}})$	Return $Dec(pk, Cmt \| Ch \| Rsp)$
Return $Cmt \| Rsp$	

The following is a special case of Lemma 3.5 of Abdalla et al. [AABN02] for seed length $s(k) = 0$. It relates the security of $\mathcal{SS}$ to that of the underlying identification scheme.

Theorem 3.4 (Security of fs-I-2-S) *Let $\mathcal{SI}$ be a non-trivial canonical standard identification scheme with commitment set* CmtSet*, and let $\mathcal{SS} = \mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{SI})$ be the associated signature scheme as per Construction 3.3. Then $\mathcal{SS}$ is unforgeable under chosen-message attack (*uf-cma*) in the random oracle model if $\mathcal{SI}$ is secure against impersonation under passive attack (*imp-pa*). Moreover, if F is a forger attacking $\mathcal{SS}$ using $\mathbf{Q}_{\mathsf{F}}^{\mathrm{Sign}}$ sign-oracle queries and $\mathbf{Q}_{\mathsf{F}}^{\mathrm{H}}$ queries to the random oracle, then there exists an impersonator A attacking $\mathcal{SI}$ such that*

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{F}}(k) \;\le\; (1 + \mathbf{Q}_{\mathsf{F}}^{\mathrm{H}})\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{SI},\mathsf{A}}(k) + \frac{(1 + \mathbf{Q}_{\mathsf{F}}^{\mathrm{H}} + \mathbf{Q}_{\mathsf{F}}^{\mathrm{Sign}})\mathbf{Q}_{\mathsf{F}}^{\mathrm{Sign}}}{2^{\beta(k)}} \;. \quad (3.3)$$

Numerical interpretation. It may be useful to elaborate on the meaning of reduction equations such as Equation (3.3). Since the number of oracle queries of a polynomial-time forger F is also polynomial, it is clear from Equation (3.3) that any forger F with non-negligible advantage in breaking $\mathcal{SS}$ gives rise to an impersonator A with a possibly much smaller but still non-negligible advantage in breaking $\mathcal{SI}$. Although this is an important asymptotical result, practitioners may be more interested in concrete security claims as pointed out in Section 2.2. From a quantitative point of view however, the guarantee offered by the above theorem is not as strong as one might hope. Suppose we allow the forger to query the hash oracle $\mathbf{Q}_{\mathsf{F}}^{\mathrm{H}} = 2^{60}$ times, and to see $\mathbf{Q}_{\mathsf{F}}^{\mathrm{Sign}} = 2^{30}$ signatures for a scheme with commitment length $\beta(k) = 160$, then even if the probability of breaking the SI scheme is estimated at 2^{-61}, all Equation (3.3) says is that F's forging probability is smaller than about $1/2$, which is not good enough.

The security lost through the transform can be compensated by using longer keys, at the expense however of an increased computational overhead. By filling in the advantage functions in Equation (3.3), we can compute how much longer the keys of the SS scheme would have to be in order to obtain the same level of security. For example, for an exponentially hard to break SI scheme, meaning that $\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{SI},\mathsf{A}}(k) = O(2^{-k})$, the keys of the SS scheme will have to be

60 bits longer than those of the SI scheme. If however the only way to break the SI scheme is by factoring a modulus of length k, we can take the advantage function to be inversely proportional to the time complexity of the fastest known factoring algorithm, which is $O(e^{1.923|N|^{1/3}\ln(|N|)^{2/3}})$ [LL93]. Putting this in Equation (3.3) yields 3102 bits as the modulus length needed to achieve the same security as the 1024-bit SI scheme. If the scheme involves cubic operation such as modular exponentiations, then this means that the computational overhead is multiplied by almost a factor 28.

These numbers however do not imply that asymptotical security is worthless for practical purposes. Schemes with asymptotical security proofs are still highly preferable over completely ad-hoc schemes, even if the reductions are very "loose". Moreover, having a proof for 3102-bit SS schemes does not necessarily mean that we can actually attack the scheme when it is instantiated with only 1024-bit moduli; a tighter proof might very well exist, maybe it just hasn't been found yet. (The lack of actual attacks is also the reason that tightness of reduction is rarely taken into account when deciding on appropriate key sizes in practice.) In this thesis we focus on asymptotical security as a first order concern; second-order concerns such as tightness of reduction are definitely important issues that are worth to be addressed, but do not form the subject of this thesis. (See also the future work suggestions in Section 5.3.)

3.4.2 Convertible Schemes and Our Transforms

In analogy with the definition of trapdoor signature schemes [DKXY03], we define the concept of *convertible identification schemes* and show how to transform these into IBI schemes. We use a slightly more general concept than the trapdoor one-way permutations used by Dodis et al. [DKXY03] that we will call *trapdoor samplable relations*. A relation $\mathbf{R}$ is a set of ordered pairs $(x, y) \in \mathrm{Dom}(\mathbf{R}) \times \mathrm{Ran}(\mathbf{R})$. We write the set of images of $x \in \mathrm{Dom}(\mathbf{R})$ as $\mathbf{R}(x) = \{y \mid (x, y) \in \mathbf{R}\}$ and the set of inverses of $y \in \mathrm{Ran}(\mathbf{R})$ as $\mathbf{R}^{-1}(y) = \{x \mid (x, y) \in \mathbf{R}\}$.

Definition 3.5 (Trapdoor Samplable Relations) A family of *trapdoor samplable relations* $\mathcal{F}$ is a triplet of polynomial-time algorithms $(\mathsf{TDG}, \mathsf{Sample}, \mathsf{Inv})$ such that the following properties hold:

- *Efficient generation:* On input 1^k, where $k \in \mathbb{N}$ is the security parameter, TDG outputs the description $\langle \mathbf{R} \rangle$ of a relation $\mathbf{R}$ in the family together with its trapdoor information t;

- *Samplability:* The output of the algorithm Sample on an input $\langle \mathbf{R} \rangle$ is a pair (x, y) uniformly distributed over $\mathbf{R}$;

- *Inversion:* On input a relation description $\langle \mathbf{R} \rangle$, the corresponding trapdoor t, and an element $y \in \mathrm{Ran}(\mathbf{R})$, algorithm Inv outputs a random element of $\mathbf{R}^{-1}(y)$;
- *Regularity:* Every relation $\mathbf{R}$ in the family is regular, meaning that the number of inverses $|\mathbf{R}^{-1}(y)|$ is the same for all $y \in \mathrm{Ran}(\mathbf{R})$.

∎

Note that this definition does not ask that any computational problem relating to the family be hard. (For example, there is no "one-wayness" requirement.) We do not need any such assumption. However, the assumed security of a cSI scheme based on the family (as will be introduced shortly) implies one-wayness of the family in the sense that computing an x such that $\mathbf{R}(x, y)$ holds, given $\langle \mathbf{R} \rangle, y$ with y drawn at random, is hard without knowing the matching trapdoor.

We also note that the relations are not required to be computable, meaning that there does not have to exist a polynomial-time algorithm that returns a member of $\mathbf{R}(x)$ on inputs $\langle \mathbf{R} \rangle, x$. In the examples we will see, such an algorithm sometimes exists and sometimes does not (e.g. for the pairing-based schemes).

Definition 3.6 (Convertible SI Schemes) A SI scheme $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ is said to be *convertible* if there exists a family of trapdoor samplable relations $\mathcal{F} = (\mathsf{TDG}, \mathsf{Sample}, \mathsf{Inv})$ such that for all $k \in \mathbb{N}$ the output of the following is distributed identically to the output of $\mathsf{Kg}(1^k)$:

$(\langle \mathbf{R} \rangle, t) \xleftarrow{R} \mathsf{TDG}(1^k)$
$(x, y) \xleftarrow{R} \mathsf{Sample}(\langle \mathbf{R} \rangle)$
$pk \leftarrow (\langle \mathbf{R} \rangle, y)$; $sk \leftarrow (\langle \mathbf{R} \rangle, x)$
Return (pk, sk)

∎

The following describes the cSI-2-IBI transform of a convertible SI (cSI) scheme into an IBI scheme. The idea is that to each identity I we can associate a value that is derivable from the master public key and I. This value plays the role of a public key for the underlying cSI scheme. This "pseudo public key" is $(\langle \mathbf{R} \rangle, \mathrm{H}(I))$, where H is a random oracle.

Construction 3.7 (The cSI-2-IBI Transform) Let $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ be a cSI scheme, and let $\mathcal{F} = (\mathsf{TDG}, \mathsf{Sample}, \mathsf{Inv})$ be the family of trapdoor samplable relations that underlies it as per Definition 3.6. The cSI-2-IBI transform associates to $\mathcal{SI}$ the random-oracle model IBI scheme $\mathcal{IBI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$ whose components we now describe. The master and user key generation algorithms are defined as

Algorithm $\mathsf{MKg}(1^k)$	Algorithm $\mathsf{UKg}(msk, I : \mathrm{H})$
$(\langle\mathbf{R}\rangle, t) \xleftarrow{R} \mathsf{TDG}(1^k)$	Parse msk as $(\langle\mathbf{R}\rangle, t)$
$mpk \leftarrow \langle\mathbf{R}\rangle$	$x \xleftarrow{R} \mathsf{Inv}(\langle\mathbf{R}\rangle, t, \mathrm{H}(I))$
$msk \leftarrow (\langle\mathbf{R}\rangle, t)$	$usk \leftarrow (\langle\mathbf{R}\rangle, x)$
Return (mpk, msk)	Return usk

where $\mathrm{H} : \{0,1\}^* \to \mathrm{Ran}(\mathbf{R})$ is a random oracle. The prover algorithm $\overline{\mathsf{P}}$ is identical to P. The verifier algorithm $\overline{\mathsf{V}}(\cdot,\cdot : \mathrm{H})$ parses its initial state as $(\langle\mathbf{R}\rangle, I)$ and runs V on initial state $(\langle\mathbf{R}\rangle, \mathrm{H}(I))$. ▮

The following theorem says that cSI-2-IBI is security-preserving.

Theorem 3.8 (Security of cSI-2-IBI) *Let $\mathcal{SI}$ be a cSI scheme and let $I\mathcal{BI} = \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{SI})$ be the associated IBI scheme as per Construction 3.7. For any* $\mathrm{atk} \in \{\mathrm{pa}, \mathrm{aa}, \mathrm{ca}\}$*, if $\mathcal{SI}$ is* imp-atk *secure then $I\mathcal{BI}$ is* imp-atk *secure.* ▮

Proof: Given any adversary $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ mounting an imp-atk attack on $I\mathcal{BI}$, we show that there exists an adversary $\mathsf{A} = (\mathsf{CV}, \mathsf{CP})$ mounting an imp-atk attack on $\mathcal{SI}$ such that

$$\mathbf{Adv}^{\text{imp-atk}}_{I\mathcal{BI},\overline{\mathsf{A}}}(k) \leq (1 + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{CV}}}) \cdot \mathbf{Adv}^{\text{imp-atk}}_{\mathcal{SI},\mathsf{A}}(k) \tag{3.4}$$

and where the algorithms CV and CP have running times $\mathbf{T}_{\mathsf{CV}} = O(\mathbf{T}_{\overline{\mathsf{CV}}} + \mathbf{Q}_{\overline{\mathsf{CV}}})$ and $\mathbf{T}_{\mathsf{CP}} = O(\mathbf{T}_{\overline{\mathsf{CP}}} + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{CP}}})$, respectively. The theorem follows.

We first sketch the algorithm, and then provide a detailed description. The CV algorithm gets a public key $pk = (\langle\mathbf{R}\rangle, y)$ as input and has access to either a Conv oracle (passive attack) or a Prov oracle (active and concurrent attack). Its strategy is to guess in advance the identity J that $\overline{\mathsf{CV}}$ will try to attack and simulate the environment of $\overline{\mathsf{CV}}$ such that it can use $\overline{\mathsf{CP}}$ to its own advantage in the second stage of the experiment. Guessing an identity from the infinite set $\{0,1\}^*$ is of course infeasible. Instead, we assume that $\overline{\mathsf{CV}}$ queries its hash oracle on J before passing control to $\overline{\mathsf{CP}}$ (if it doesn't, we let CV query $\mathrm{H}(J)$ itself), so that CV can guess the index $q_{\text{guess}} \xleftarrow{R} \{1 \ldots \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{CV}}} + 1\}$ of the *crucial hash query* in advance. It then runs $\overline{\mathsf{CV}}$ on input $mpk = \langle\mathbf{R}\rangle$, simulating $\overline{\mathsf{CV}}$'s oracle queries as follows (we ignore the bookkeeping of honest and corrupted identities here):

- $\mathrm{H}(I)$: If I was queried before, return the same value as before. If this is the q_{guess}th unique hash query, then let $I_{\text{guess}} \leftarrow I$ and return y (that was part of CV's input) as the hash value. Otherwise, return the second part of a pair generated by the Sample algorithm as the hash value, and keep the first part as the corresponding user secret key.
- Init(I): Run $\mathrm{H}(I)$ and return 1.

– $\textsc{Conv}(I)$ (or $\textsc{Prov}(I, s)$): If $I = I_{\text{guess}}$, forward a conversation (response) generated by CV's own conversation (prover) oracle, otherwise simulate a real conversation (prover) using the user secret key stored during the first $\mathrm{H}(I)$ query.

– $\textsc{Corr}(I)$: If $I = I_{\text{guess}}$, abort. Otherwise, return the user secret key corresponding to I.

until $\overline{\mathsf{CV}}$ returns the identity to be impersonated J and state information for $\overline{\mathsf{CP}}$. If $J \neq I_{\text{guess}}$, meaning that CV's guess was wrong, it aborts. If $J = I_{\text{guess}}$ however, CV proceeds to the second phase of the game and transfers control to the cheating prover CP. On any incoming message M_{in}, CP simply forwards $\overline{\mathsf{CP}}$'s reply to the same message, answering $\overline{\mathsf{CP}}$'s oracle queries in the same way as CV did before, but returning $\perp$ on $\textsc{Conv}$, $\textsc{Prov}$ and $\textsc{Corr}$ requests for J.

A detailed description of algorithms CV, CP is given below. Given an adversary $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ mounting an imp-atk attack on $\mathcal{IBI}$, we construct an adversary $\mathsf{A} = (\mathsf{CV}, \mathsf{CP})$ mounting an imp-atk attack on $\mathcal{SI}$ as follows (the subroutines used to simulate $\overline{\mathsf{CV}}$'s and $\overline{\mathsf{CP}}$'s oracles are given in Figure 3.6):

Algorithm $\mathsf{CV}(pk : \textsc{Or})$ // $\textsc{Or}$ *is a conversation oracle if* atk = pa
  Parse pk as $(\langle \mathbf{R} \rangle, y)$ // *and a prover oracle if* atk $\in \{\text{aa}, \text{ca}\}$
  $mpk \leftarrow \langle \mathbf{R} \rangle$
  $HU \leftarrow \emptyset\,;\ CU \leftarrow \emptyset\,;\ QU \leftarrow \emptyset$ // *honest, corrupted and queried users*
  $PID \leftarrow \emptyset$ // *set of running prover sessions*
  $q_{\text{guess}} \xleftarrow{R} \{1, \ldots, \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{CV}}} + 1\}$ // *guess for crucial hash query*
  If atk = pa then $(J, St_{\overline{\mathsf{CP}}}) \xleftarrow{R} \overline{\mathsf{CV}}(mpk : \mathsf{Init\text{-}sim}, \mathsf{Corr\text{-}sim}, \mathsf{Conv\text{-}sim}, \mathsf{H\text{-}sim})$
  Else $(J, St_{\overline{\mathsf{CP}}}) \xleftarrow{R} \overline{\mathsf{CV}}(mpk : \mathsf{Init\text{-}sim}, \mathsf{Corr\text{-}sim}, \mathsf{Prov\text{-}sim}, \mathsf{H\text{-}sim})$
  If $|QU| < q_{\text{guess}}$ or $J \neq I_{\text{guess}}$ then abort
  $HU \leftarrow HU \setminus \{J\}\,;\ CU \leftarrow CU \cup \{J\}$
  $St_{\mathsf{CP}} \leftarrow (St_{\overline{\mathsf{CP}}}, \langle \mathbf{R} \rangle, HU, CU, QU, HTbl, UTbl, I_{\text{guess}}, q_{\text{guess}})$
    // *tables HTbl and UTbl contain hash values and user secret keys*
  Return (J, St_{CP})

Algorithm $\mathsf{CP}(M_{\text{in}}, St_{\mathsf{CP}})$
  Parse St_{CP} as $(St_{\overline{\mathsf{CP}}}, \langle \mathbf{R} \rangle, HU, CU, QU, HTbl, UTbl, I_{\text{guess}}, q_{\text{guess}})$
  If atk = pa then
    $(M_{\text{out}}, St_{\overline{\mathsf{CP}}}) \xleftarrow{R} \overline{\mathsf{CP}}(M_{\text{in}}, St_{\overline{\mathsf{CP}}} : \mathsf{Init\text{-}sim}, \mathsf{Corr\text{-}sim}, \mathsf{Conv\text{-}sim}, \mathsf{H\text{-}sim})$
  Else $(M_{\text{out}}, St_{\overline{\mathsf{CP}}}) \xleftarrow{R} \overline{\mathsf{CP}}(M_{\text{in}}, St_{\overline{\mathsf{CP}}} : \mathsf{Init\text{-}sim}, \mathsf{Corr\text{-}sim}, \mathsf{Prov\text{-}sim}, \mathsf{H\text{-}sim})$
  $St_{\mathsf{CP}} \leftarrow (St_{\overline{\mathsf{CP}}}, \langle \mathbf{R} \rangle, HU, CU, QU, HTbl, UTbl, I_{\text{guess}}, q_{\text{guess}})$
  Return $(M_{\text{out}}, St_{\mathsf{CP}})$

Let us explain why CV and CP create a perfect simulation of $\overline{\mathsf{A}}$'s environment. The input of $\overline{\mathsf{CV}}$ is correctly distributed because by Definition 3.6 and

Subroutine $\mathsf{Init\text{-}sim}(I)$
  If $I \in CU \cup HU$ then return $\perp$
  $temp \leftarrow \mathsf{H\text{-}sim}(I)$
  $HU \leftarrow HU \cup \{I\}$
  Return 1

Subroutine $\mathsf{Corr\text{-}sim}(I)$
  If $I \notin HU$ then return $\perp$
  $CU \leftarrow CU \cup \{I\}$
  $HU \leftarrow HU \setminus \{I\}$
  If $I = I_{\text{guess}}$ then abort
  Return $(\langle \mathbf{R} \rangle, UTbl[I])$

Subroutine $\mathsf{Conv\text{-}sim}(I)$
  If $I \notin HU$ then return $\perp$
  If $I = I_{\text{guess}}$ then $C \leftarrow \text{OR}(\varepsilon)$
  Else
    $R_{\overline{\mathsf{P}}} \xleftarrow{R} \{0,1\}^{\rho_{\overline{\mathsf{P}}}}$; $R_{\overline{\mathsf{V}}} \xleftarrow{R} \{0,1\}^{\rho_{\overline{\mathsf{V}}}}$
    $St_{\overline{\mathsf{P}}} \leftarrow (\langle \mathbf{R} \rangle, UTbl[I])$
    $St_{\overline{\mathsf{V}}} \leftarrow (\langle \mathbf{R} \rangle, HTbl[I])$
    $M_{\text{in}} \leftarrow \varepsilon$; $C \leftarrow \varepsilon$
    While $St_{\overline{\mathsf{V}}} \notin \{\mathtt{acc}, \mathtt{rej}\}$ do
      $(M_{\text{out}}, St_{\overline{\mathsf{P}}}) \leftarrow \overline{\mathsf{P}}(M_{\text{in}}, St_{\overline{\mathsf{P}}}, R_{\overline{\mathsf{P}}})$
      $(M_{\text{in}}, St_{\overline{\mathsf{V}}}) \leftarrow \overline{\mathsf{V}}(M_{\text{out}}, St_{\overline{\mathsf{V}}}, R_{\overline{\mathsf{V}}})$
      $C \leftarrow C \| M_{\text{out}} \| M_{\text{in}}$
  Return C

Subroutine $\mathsf{H\text{-}sim}(I)$
  If $I \notin QU$ then
    $QU \leftarrow QU \cup \{I\}$
    If $|QU| = q_{\text{guess}}$ then
      $I_{\text{guess}} \leftarrow I$; $HTbl[I] \leftarrow y$
    Else $(UTbl[I], HTbl[I])$
      $\xleftarrow{R} \mathsf{Sample}(\langle \mathbf{R} \rangle)$
  Return $HTbl[I]$

Subroutine $\mathsf{Prov\text{-}sim}(I, s, M_{\text{in}})$
  If $I \notin HU$ then return $\perp$
  If $I = I_{\text{guess}}$ then
    $M_{\text{out}} \leftarrow \text{OR}(s, M_{\text{in}})$
  Else
    If $(I, s) \notin PID$ then
      If atk = aa then
        $PID \leftarrow \{(I, s)\}$
      If atk = ca then
        $PID \leftarrow PID \cup \{(I, s)\}$
      $R_{\overline{\mathsf{P}}} \xleftarrow{R} \{0,1\}^{\rho_{\overline{\mathsf{P}}}}$
      $St_{\overline{\mathsf{P}}}[I, s] \leftarrow (\langle \mathbf{R} \rangle, UTbl[I], R_{\overline{\mathsf{P}}})$
    $(M_{\text{out}}, St_{\overline{\mathsf{P}}}[I, s]) \leftarrow$
      $\overline{\mathsf{P}}(M_{\text{in}}, St_{\overline{\mathsf{P}}}[I, s] : \mathsf{H\text{-}sim})$
  Return M_{out}

Figure 3.6: Subroutines of CV and CP used to simulate $\overline{\mathsf{CV}}$'s and $\overline{\mathsf{CP}}$'s oracles in the proof of Theorem 3.8.

Construction 3.7 both the relation description included in the public key pk of a cSI scheme and the relation description included in the master public key mpk of its $\mathsf{cSI\text{-}2\text{-}IBI}$ transform are generated by the same algorithm $\mathsf{TDG}(1^k)$.

The regularity of $\mathbf{R}$ and the uniform distribution of the output of the Sample algorithm over $\mathbf{R}$ together imply that $\mathsf{H\text{-}sim}(I) = HTbl[I]$ is uniformly distributed over $\text{Ran}(\mathbf{R})$ and that $UTbl[I]$ is uniformly distributed over the inverses of the entries of $HTbl[I]$, exactly like the outputs of a real random oracle and the UKg algorithm. The correct distribution of the entries of $UTbl$ as user secret keys for all $I \neq I_{\text{guess}}$ also insures that the replies of the CONV, PROV and CORR oracles are identically distributed as those in a real attack on $I\mathcal{BI}$. The same is true for the CONV and PROV oracles for $I = I_{\text{guess}}$, because the secret

key sk underlying A's conversation or prover oracle is also uniformly distributed over $\mathbf{R}^{-1}(\mathsf{H\text{-}sim}(I))$ by Definition 3.6.

The only possible whistle-blower is the CORR oracle, which gives up when queried on $I = I_{\text{guess}}$ instead of returning the corresponding user secret key. The occurrence of such query, however, means that A was bound to lose the game anyway, since the identity impersonated by $\overline{\mathsf{CP}}$ must be an uncorrupted identity and can under no circumstances be equal to I_{guess} anymore. So in the cases that matter to A, the query will simply not occur.

The initial state of $\overline{\mathsf{CP}}$ is generated by $\overline{\mathsf{CV}}$ as it is supposed to, and $\overline{\mathsf{CP}}$'s incoming protocol messages are correctly distributed because by Construction 3.7 $\overline{\mathsf{V}}$ runs V as a subroutine. The replies from $\overline{\mathsf{CP}}$'s oracles are identical to those in a real attack for the same reasons.

So conditioned on the event that $J = I_{\text{guess}}$, the simulation of $\overline{\mathsf{A}}$'s environment is perfect. This means that A's impersonation is successful whenever $J = I_{\text{guess}}$ and $\overline{\mathsf{A}}$ succeeds, and that A's advantage can be lower bounded as

$$\begin{aligned}
\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{SI},\mathsf{A}}(k) &\geq \Pr\left[J = I_{\text{guess}} \wedge \mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) = 1\right] \\
&= \Pr\left[J = I_{\text{guess}}\right] \cdot \Pr\left[\mathbf{Exp}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) = 1\right] \\
&\geq \frac{1}{1 + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{CV}}}} \cdot \mathbf{Adv}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k)
\end{aligned}$$

where the first equality holds because the simulation of $\overline{\mathsf{A}}$'s environment is perfect and hence $J = I_{\text{guess}}$ and $\overline{\mathsf{A}}$'s success are unrelated events. The claims for the running times of CV and CP can easily be verified from the descriptions of the algorithms. ∎

Convertibility of a standard signature (SS) scheme $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ is defined by analogy to Definition 3.6. (The condition is only on the key-generation algorithm.) The cSS-2-IBS transform is defined analogously to the cSI-2-IBI transform:

Construction 3.9 (The cSS-2-IBS Transform) To any convertible SS (cSS) scheme $\mathcal{SS} = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$, the cSS-2-IBS transform associates an IBS scheme $\mathcal{IBS} = \mathsf{cSS\text{-}2\text{-}IBS}(\mathcal{SS}) = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{Sign}}, \overline{\mathsf{Vf}})$ where the master and the user key generators are exactly as in Construction 3.7, and where $\overline{\mathsf{Sign}}(usk, \cdot)$ and $\overline{\mathsf{Vf}}(mpk, I, \cdot, \cdot : \mathrm{H})$ are implemented as $\mathsf{Sign}(usk, \cdot)$ and $\mathsf{Vf}((mpk, \mathrm{H}(I)), \cdot, \cdot)$, respectively. ∎

The proof of the following analogue of Theorem 3.8 showing that the cSS-2-IBS transform is security preserving is similar to the proof of Theorem 3.8 and is thus omitted.

Theorem 3.10 (Security of cSS-2-IBS) *Let $\mathcal{SS}$ be a cSS scheme and let $\mathcal{IBS}$ = cSS-2-IBS($\mathcal{SS}$) be the associated IBS scheme as defined in Construction 3.9. If $\mathcal{SS}$ is* uf-cma *secure then $\mathcal{IBS}$ is also* uf-cma *secure. Moreover, for any adversary $\overline{\mathsf{F}}$ attacking $\mathcal{IBS}$, there exists an adversary F attacking $\mathcal{SS}$ such that*

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{IBS},\overline{\mathsf{F}}}(k) \leq (1 + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{F}}}) \cdot \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{F}}(k) \tag{3.5}$$

and where F runs in time $O(\mathbf{T}_{\overline{\mathsf{F}}} + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{F}}})$. ∎

One can check that any trapdoor SS (tSS) scheme [DKXY03] is a cSS scheme, and that their tSS-2-IBS transform coincides with cSS-2-IBS in case the starting cSS scheme is trapdoor. Thus, Theorem 3.10 represents a (slight) extension of their result. However, the extension is important, for we will see cases of cSS schemes that are not trapdoor and where the extension is needed.

By Theorem 3.4 we know that if $\mathcal{SI}$ is an imp-pa secure SI scheme, then fs-I-2-S($\mathcal{SI}$) is a uf-cma secure SS scheme [AABN02]. Because the convertibility property only imposes requirements on the key generation algorithm, which is left untouched by the fs-I-2-S transform, one can also see that the fs-I-2-S transform of a canonical cSI scheme is a cSS scheme. Combining this with Theorem 3.10 yields the following, which will be our main tool to prove security of IBS schemes. All practical SI and IBI schemes considered in this paper are canonical.

Corollary 3.11 *Let $\mathcal{SI}$ be a non-trivial canonical cSI scheme, and let $\mathcal{IBS}$ =* cSS-2-IBS(fs-I-2-S($\mathcal{SI}$)) *as per Construction 3.9. If $\mathcal{SI}$ is* imp-pa *secure then $\mathcal{IBS}$ is* uf-cma *secure.* ∎

The above corollary gives a purely asymptotical security claim for the IBS scheme. By combining Equations (3.3) and (3.5), one can see that the concrete security reduction from the IBS to the cSI scheme is particularly loose due to the multiplication of the factors $\mathbf{Q}^{\mathrm{H}}_{\mathsf{F}}$ in Equation (3.3) and $\mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{F}}}$ in Equation (3.5). Theoretically speaking, this means that a factoring-based IBS scheme, when faced with an adversary capable of doing 2^{60} queries to both random oracles, needs to use 6701-bit moduli to achieve the same level of security as a 1024-bit SI scheme.

The canonicity definition for SI schemes is easily extended to IBI schemes, the only modification being that the verifier's decision is a deterministic function $Dec((mpk, I), Cmt \| Ch \| Rsp)$ of the master public key mpk, the user's identity I and the communication transcript. One can apply the fs-I-2-S transform to a canonical IBI scheme to obtain an IBS scheme, and one can check that cSS-2-IBS(fs-I-2-S($\mathcal{SI}$)) = fs-I-2-S(cSI-2-IBI($\mathcal{SI}$)) for any canonical cSI scheme $\mathcal{SI}$. It follows that fs-I-2-S yields a uf-cma secure IBS scheme if it is applied to a *converted* IBI scheme, meaning one that is obtained as the result of applying

cSI-2-IBI to some (canonical) cSI scheme. However, one can also apply fs-I-2-S to a canonical IBI scheme that is not converted and get an IBS scheme, and there will be instances later where we would like to do this. Unfortunately, the IBS scheme so obtained need not be secure, in the sense that the analogue of the result of Theorem 3.4 does not hold, as shown by the following proposition.

Proposition 3.12 Assume there exists an imp-pa secure canonical IBI scheme. Then, there exists an imp-pa secure canonical IBI scheme $\mathcal{IBI}$ such that $\mathcal{IBS} =$ fs-I-2-S($\mathcal{IBI}$) is not uf-cma secure. ∎

Proof: Let $\mathcal{IBI}$ be identical to the given imp-pa secure IBI scheme, except that the decision function is relaxed so that the verifier also accepts when the challenge is equal to the identity being verified. Since an imp-pa adversary has to commit to an identity J before the challenge is drawn, and since the latter is drawn from a set of super-polynomial size (this follows from the assumed imp-pa security of the original IBI scheme), this change will not affect the security of the IBI scheme, meaning $\mathcal{IBI}$ is also imp-pa secure. However the corresponding IBS scheme $\mathcal{IBS} =$ fs-I-2-S($\mathcal{IBI}$) is insecure since a tuple (Cmt, Rsp) is a valid signature for message M under identity $I = \mathrm{H}(Cmt \| M)$. ∎

We now provide a remedy for the above. We consider the *extended Fiat-Shamir transform* efs-IBI-2-IBS, a modified version of the fs-I-2-S transform that hashes the identity of the signer (prover) along with the commitment and message, rather than merely hashing the commitment and message as in fs-I-2-S. We show (by an extension of the proof of Abdalla et al. [AABN02] that, if this transform is applied to a canonical imp-pa secure IBI scheme, then the outcome is a uf-cma secure IBS scheme. We apply this in Section 3.6 to obtain uf-cma secure IBS schemes from the two unconverted IBI schemes we consider, namely $\mathcal{OKDL}$-$\mathcal{IBI}$ and $\mathcal{XDL}$-$\mathcal{IBI}$.

Construction 3.13 (The efs-IBI-2-IBS Transform) Let $\mathcal{IBI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$ be a canonical IBI scheme with commitment set function CmtSet($\cdot$), challenge set ChSet($\cdot$) and decision function $Dec(\cdot, \cdot, \cdot)$. The *extended Fiat-Shamir transform* efs-IBI-2-IBS associates to $\mathcal{IBI}$ an IBS scheme $\mathcal{IBS} =$ efs-IBI-2-IBS($\mathcal{IBI}$) $= (\mathsf{MKg}, \mathsf{UKg}', \overline{\mathsf{Sign}}, \overline{\mathsf{Vf}})$ defined as:

Algorithm $\mathsf{UKg}'(msk, I)$
$usk \leftarrow \mathsf{UKg}(msk, I)$
$usk' \leftarrow (usk, I)$
Return usk'

Algorithm $\overline{\mathsf{Vf}}(mpk, I, M, \sigma : \mathrm{H})$
Parse σ as $Cmt \| Rsp$
$Ch \leftarrow \mathrm{H}(I \| Cmt \| M)$
Return $Dec(mpk, I, Cmt \| Ch \| Rsp)$

Algorithm $\overline{\mathsf{Sign}}(usk', M : \mathrm{H})$
Parse usk' as (usk, I)
$(Cmt, St_{\overline{\mathsf{P}}}) \xleftarrow{R} \overline{\mathsf{P}}(\varepsilon, usk)$
$Ch \leftarrow \mathrm{H}(I \| Cmt \| M)$
$(Rsp, St_{\overline{\mathsf{P}}}) \xleftarrow{R} \overline{\mathsf{P}}(Ch, St_{\overline{\mathsf{P}}})$
Return $Cmt \| Rsp$

where $\mathrm{H} : \{0,1\}^* \to \mathrm{ChSet}$ is a random oracle. ∎

The following theorem states the result under this transform. It implies that, given a canonical three-move IBI scheme secure under passive attacks, the corresponding IBS scheme under the extended FS transform is unforgeable under adaptive chosen-message attacks in the random oracle model, assuming that the commitment space is large enough.

Theorem 3.14 (Security of $\mathsf{efs\text{-}IBI\text{-}2\text{-}IBS}$) *Let $\mathcal{IBI}$ be a non-trivial canonical IBI scheme with commitment length $\beta(k)$, and let $\mathcal{IBS} = \mathsf{efs\text{-}IBI\text{-}2\text{-}IBS}(\mathcal{IBI})$ be the corresponding IBS scheme as per the extended Fiat-Shamir transform of Construction 3.13. If $\mathcal{IBI}$ is polynomially secure against impersonation under passive attack, then $\mathcal{IBS}$ is polynomially unforgeable under chosen-message attack in the random oracle model. Moreover, if $\overline{\mathsf{F}}$ is a forger attacking $\mathcal{IBS}$ using $\mathbf{Q}^{\text{Sign}}_{\overline{\mathsf{F}}}$ sign-oracle queries and $\mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{F}}}$ queries to the random oracle, then there exists a passive impersonator $\overline{\mathsf{A}}$ attacking $\mathcal{IBI}$ such that*

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{IBS},\overline{\mathsf{F}}}(k) \;\le\; (1 + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{F}}}) \cdot \mathbf{Adv}^{\text{imp-pa}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k) + \frac{(1 + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{F}}} + \mathbf{Q}^{\text{Sign}}_{\overline{\mathsf{F}}}) \cdot \mathbf{Q}^{\text{Sign}}_{\overline{\mathsf{F}}}}{2^{\beta(k)}}$$

where $\overline{\mathsf{A}}$'s running time is equal to $O(\mathbf{T}_{\overline{\mathsf{F}}} + \mathbf{Q}_{\overline{\mathsf{F}}})$ while the number of conversation queries of $\overline{\mathsf{A}}$ is at most $2 \cdot \mathbf{Q}^{\text{Sign}}_{\overline{\mathsf{F}}}$. ∎

Proof: The proof of Theorem 3.14 follows a standard approach. Given $\overline{\mathsf{F}}$ attacking $\mathcal{IBS}$, we construct $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ attacking $\mathcal{IBI}$ by running $\overline{\mathsf{F}}$ properly and using the forgery to impersonate the prover. First, $\overline{\mathsf{CV}}$ guesses the index of the hash query $I \| Cmt \| M$ that will be involved in the forgery. We call this hash query the *crucial hash query*. The adversary $\overline{\mathsf{CV}}$ then runs $\overline{\mathsf{F}}$ using its Init oracle to answer $\overline{\mathsf{F}}$'s Init queries, its Corr oracle to answer $\overline{\mathsf{F}}$'s Corr queries, and its Conv oracle to answer $\overline{\mathsf{F}}$'s random oracle and Sign queries (using the challenges in the conversations as hash values). Upon receiving the crucial hash query, $\overline{\mathsf{CV}}$ announces that it will impersonate $J = I$ and transfers control to $\overline{\mathsf{CP}}$. $\overline{\mathsf{CP}}$ then sends Cmt to $\overline{\mathsf{V}}$ to obtain Ch and returns Ch as the answer to

$\overline{\mathsf{F}}$'s (crucial) hash query. Assuming that $\overline{\mathsf{CV}}$ guesses correctly, when $\overline{\mathsf{F}}$ outputs a forgery $\sigma = Cmt \| Rsp$ for message M and identity J, the adversary $\overline{\mathsf{CP}}$ sends Rsp to $\overline{\mathsf{V}}$, thus completing the impersonation. Similar to the previous phase, if $\overline{\mathsf{F}}$ makes any queries during this phase, $\overline{\mathsf{CP}}$ replies using its oracle access as did $\overline{\mathsf{CV}}$ discussed above.
The proof closely resembles that of Theorem 3.4 [AABN02]. We omit details and make the following remarks. First, we point out that, as dictated in the experiment $\mathbf{Exp}^{\text{imp-pa}}_{\mathit{IBI},\overline{\mathsf{A}}}(k)$, the adversary $\overline{\mathsf{CP}}$ is not allowed to submit queries involving the identity J to the oracle Conv. However, it is possible that $\overline{\mathsf{F}}$ makes Sign and H queries involving J during $\overline{\mathsf{CP}}$'s simulation. We can get around this problem by having $\overline{\mathsf{CV}}$ query the Conv oracle for a batch of $\mathbf{Q}^{\text{Sign}}_{\overline{\mathsf{F}}}$ transcripts for identity J right before transferring control to $\overline{\mathsf{CP}}$, resulting in the doubling of $\overline{\mathsf{A}}$'s number of conversation oracle queries.
Finally, we point out that the proof here does not work for the fs-I-2-S transform because $\overline{\mathsf{CV}}$ needs to announce the identity to impersonate J before transferring control to $\overline{\mathsf{CP}}$. Under the efs-IBI-2-IBS transform, the identity is part of the crucial hash query and hence $\overline{\mathsf{CV}}$ can already announce it to $\overline{\mathsf{V}}$ and return the challenge as the corresponding hash value. The same is not true under the fs-I-2-S transform however, since the identity to be forged by $\overline{\mathsf{F}}$ may not have been initiated yet. ∎

3.5 Applying the Framework

We now apply the above transform-based framework to prove security of existing and new IBI and IBS schemes. To do this, we consider numerous SI schemes. (Some are known, some are new.) We show that they are convertible, and then analyze their security. The implications for corresponding IBI and IBS schemes, obtained via the transforms discussed above, follow from Theorem 3.8 and Corollary 3.11.

Reset Lemma. When proving the security of SI schemes, we will make heavy use of Bellare and Palacio's [BP02] Reset Lemma, which limits the success probability of a cheating prover CP in any canonical identification scheme as a function of the probability of obtaining two accepting conversations in a rewinding experiment. This experiment tries to extract two correct responses to two different challenges for the same commitment from the prover. Note that by making abstraction of the verifier's initial state, the Reset Lemma can be applied to both SI and IBI schemes.

Lemma 3.15 (Reset Lemma [BP02]) Let CP be a prover in a canonical standard or identity-based identification scheme with commitment length $\beta(k)$, challenge length $\mathrm{ChSet}(\cdot)$ and decision function Dec. Let St_{V} and let St_{CP} be the

initial states of the verifier and cheating prover, respectively. Let $acc(St_{\mathsf{CP}}, St_{\mathsf{V}})$ be the probability that the verifier accepts when initiated with state St_{V} after interacting with CP initiated with St_{CP}, and let $res(St_{\mathsf{CP}}, St_{\mathsf{V}})$ be the probability that the following experiment returns 1:

$R_{\mathsf{CP}} \xleftarrow{R} \{0,1\}^{\rho_{\mathsf{CP}}}$; $(Cmt, St_{\mathsf{CP}}) \leftarrow \mathsf{CP}(\varepsilon, St_{\mathsf{CP}}, R_{\mathsf{CP}})$
$Ch_1 \xleftarrow{R} \mathrm{ChSet}(St_{\mathsf{V}})$; $(Rsp_1, St'_{\mathsf{CP}}) \leftarrow \mathsf{CP}(Ch_1, St_{\mathsf{CP}}, R_{\mathsf{CP}})$
$d_1 \leftarrow Dec(St_{\mathsf{V}}, Cmt, Ch_1, Rsp_1)$
$Ch_2 \xleftarrow{R} \mathrm{ChSet}(St_{\mathsf{V}})$; $(Rsp_2, St'_{\mathsf{CP}}) \leftarrow \mathsf{CP}(Ch_2, St_{\mathsf{CP}}, R_{\mathsf{CP}})$
$d_2 \leftarrow Dec(St_{\mathsf{V}}, Cmt, Ch_2, Rsp_2)$
If ($d_1 = 1$ and $d_2 = 1$ and $Ch_1 \neq Ch_2$) then return 1 else return 0

Then,

$$acc(St_{\mathsf{CP}}, St_{\mathsf{V}}) \leq 2^{-\beta(k)} + \sqrt{res(St_{\mathsf{CP}}, St_{\mathsf{V}})}. \tag{3.6}$$

∎

Zero-knowledge proofs. Most practical identification schemes are based on *zero-knowledge proofs*. These were introduced by Goldwasser et al. [GMR89] and first applied in identification schemes by Fiat and Shamir [FS86]. We refer to the book by Goldreich [Gol01] for more details on zero-knowledge proofs. A famous and extremely accessible introduction to zero-knowledge is given by Quisquater et al. [QQQ+90].

Loosely speaking, zero-knowledge proofs are proofs that yield nothing but the validity of a claim. More formally, a pair of interactive algorithms (P, V) is an *interactive proof system* for a language $L \subseteq \{0,1\}^*$ if V is polynomial-time and

- *Completeness:* for all $x \in L$, there exists an auxiliary input $y \in \{0,1\}^*$ such that with overwhelming probability V initialized with x accepts after interacting with P, when initialized with y.
- *Soundness:* for all $x \notin L$, all interactive algorithms CP and all $y \in \{0,1\}^*$, V accepts with negligible probability after interacting with CP when initialized with x and y, respectively.

Intuitively, an interactive proof system is said to be *zero-knowledge* if the verifier doesn't learn anything new from the interaction with the prover, meaning that anything it can compute afterwards, it could have computed before the interaction as well. We say that an interactive proof system (P, V) is *perfect zero-knowledge* if for every probabilistic polynomial-time interactive algorithm CV, called the cheating verifier, there exists a probabilistic polynomial-time (non-interactive) algorithm Sim, called the conversation simulator, such that for every $x \in L$ the output of the experiment

$T \leftarrow \varepsilon$; $M_{\text{in}} \leftarrow \varepsilon$; $St_{\mathsf{P}} \leftarrow y$; $St_{\mathsf{CV}} \leftarrow x$
Repeat
  $(M_{\text{out}}, St_{\mathsf{P}}) \stackrel{R}{\leftarrow} \mathsf{P}(M_{\text{in}}, St_{\mathsf{P}})$
  $(M_{\text{in}}, St_{\mathsf{CV}}) \stackrel{R}{\leftarrow} \mathsf{CV}(M_{\text{out}}, St_b adv)$
  $T \leftarrow T \| M_{\text{in}} \| M_{\text{out}}$
Until $St_{\mathsf{CV}} \in \{\mathtt{acc}, \mathtt{rej}\}$
Return T

is identically distributed to the output of $\mathsf{Sim}(x)$. The proof system is said to be *statistical zero-knowledge* if the statistical distance between the above distributions is negligible in $|x|$, and is *computational zero-knowledge* if no probabilistic polynomial-time can successfully distinguish between the above distributions with probability non-negligibly better than $1/2$. The weaker notion of *honest-verifier zero-knowledge* (HVZK) only requires the above conditions to hold for the honest verifier V, instead of for any possibly cheating verifier CV.

The proof systems as defined above are also referred to as *proofs of membership*, because P proves that x is a member of the language L. *Proofs of knowledge*, on the other hand, convince the verifier that the prover "knows" something. Let $\mathbf{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation, and let $L_{\mathbf{R}} = \{x \mid (x,y) \in \mathbf{R}\}$ be the language associated to $\mathbf{R}$. We say that a pair of interactive algorithms (P, V) is an *interactive proof of knowledge system* for $\mathbf{R}$ if V is a probabilistic polynomial-time algorithm and if

- *Completeness:* V initialized with x accepts with overwhelming probability after interacting with P initialized with y for all $(x,y) \in \mathbf{R}$.
- *Soundness:* there exists a probabilistic polynomial-time algorithm Ext, called the *knowledge extractor*, such that for every $x \in L_{\mathbf{R}}$ and for every polynomial-time interactive algorithm CP that has non-negligible probability of making V accept when initialized with x, Ext, on input x and a description of algorithm CP, outputs y such that $(x,y) \in \mathbf{R}$ with non-negligible probability.

Zero-knowledge proofs of knowledge are defined analogously to zero-knowledge proofs of membership through the existence of a simulator.

Proofs of knowledge can be used to build standard identification schemes by taking the first component of a tuple $(x,y) \in \mathbf{R}$ to be the public key, and the second to be the secret key. The resulting SI scheme $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ is secure under active (respectively passive) attack (imp-aa) if (P, V) is a zero-knowledge (respectively honest-verifier zero-knowledge) interactive proof of knowledge system for the relation $\mathbf{R} = [\mathsf{Kg}(1^k)]$, and if computing secret keys from public keys is hard. This is easily seen from the fact that an imp-pa/aa adversary $\mathsf{A} = (\mathsf{CV}, \mathsf{CP})$ can be turned into an algorithm that retrieves the secret key by using the Sim algorithm to simulate provers (or conversations) for CV, and using the Ext algorithm to extract the secret key from CP.

Algorithm $\mathsf{Kg}(1^k)$
$(N, p, q) \xleftarrow{R} \mathsf{K}_{\text{fact}}(1^k)$
For $i = 1 \ldots t$ do
$x_i \xleftarrow{R} \mathbb{Z}_N^*$; $X_i \leftarrow {x_i}^{-2^m} \bmod N$
$pk \leftarrow (N, (X_1, \ldots, X_t))$; $sk \leftarrow (N, (x_1, \ldots, x_t))$
Return (pk, sk)

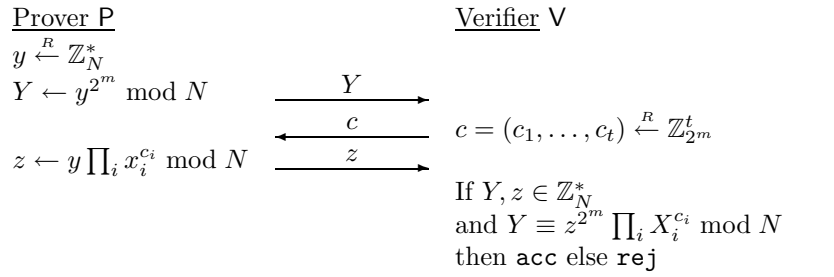


Figure 3.7: **The *ItR-SI* and *FS-SI* schemes.** The scheme is parameterized with modulus generator K_{fact}, exponent $m \geq 1$ and the number of roots t. The prover P and verifier V are run on initial states $sk = (N, (X_1, \ldots, X_t))$ and $pk = (N, (x_1, \ldots, x_t))$, respectively. The *FS-SI* scheme is the special case for $m = 1$, and the scheme of Feige, Fiat and Shamir [FFS88] is a slight variation with a Blum integer as the modulus N.

3.5.1 Schemes based on Factoring

The key generation algorithms of all factoring-based schemes are specified in terms of an abstract modulus generator K_{fact} as defined in Section 2.3.1. We make no assumptions on the modulus generator, except that the related factoring problem (given N, compute p, q) cannot be solved in poly(k) time.

The Fiat-Shamir and the iterated-root schemes. Quite remarkably, the scheme that can be considered as the mother of all identification schemes by Fiat and Shamir [FS86] was already presented as an IBI scheme, denoted here as *FS-IBI*. Later Feige et al. [FFS88] proposed a variant and proved it secure under active attacks. The scheme was further generalized to L-th roots with $\gcd(L, \varphi(n)) > 1$ by Ohta and Okamoto [OO90] and to 2^mth roots by Ong and Schnorr [OS90]. We show this scheme in Figure 3.7 and call it the *ItR-SI* (for *iterated root*) scheme. The descriptions of all schemes in this work explicitly include membership tests on the messages sent by the prover, to prevent the type of attacks described by Burmester and Desmedt [BD89] that e.g. send zero as the commitment. Two related schemes due to Micali and Shamir [MS88] and Guillou et al. [GUQ01] are, although based on the factoring assumption,

not convertible because they use fixed values for X_i instead of random squares modulo N.

Since $\mathcal{FS}$-$\mathcal{SI}$ is the special case of $It\mathcal{R}$-$\mathcal{SI}$ in which $m = 1$, it suffices to show that the latter is convertible. This is easily seen by considering the relation $\mathbf{R} = \{((x_1, \ldots, x_t), (X_1, \ldots, X_t)) \mid X_i \equiv x_i^{-2^m} \bmod N \text{ for } i = 1, \ldots, t\}$ with description $\langle \mathbf{R} \rangle = N$ and trapdoor (p, q). Pair sampling involves selecting random elements from $\mathbb{Z}_N^*$, raising them to the 2^m-th power, and inverting them modulo N.

We note that $\mathcal{FS}$-$\mathcal{IBI}$ = $\mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{FS}\text{-}\mathcal{SI})$ is exactly the IBI scheme as presented by Fiat and Shamir [FS86] and $\mathcal{FS}$-$\mathcal{IBS}$ = $\mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{FS}\text{-}\mathcal{SI}))$ is exactly their IBS scheme. We know that $\mathcal{FS}$-$\mathcal{SI}$ is imp-pa and imp-aa secure assuming factoring is hard [FFS88], and this easily extends to imp-ca. Theorem 3.8 implies that $\mathcal{FS}$-$\mathcal{IBI}$ inherits these security attributes. (Corollary 3.11 implies the uf-cma security of $\mathcal{FS}$-$\mathcal{IBS}$ assuming factoring is hard, but this was known [DKXY03].)

We know that $It\mathcal{R}$-$\mathcal{SI}$ is imp-pa and imp-aa secure assuming factoring is hard [Sho99, Sch96]. Theorem 3.8 implies that $It\mathcal{R}$-$\mathcal{IBI}$ = $\mathsf{cSI\text{-}2\text{-}IBI}(It\mathcal{R}\text{-}\mathcal{SI})$ is imp-pa and imp-aa secure assuming factoring is hard. Corollary 3.11 implies that $It\mathcal{R}$-$\mathcal{IBS}$ = $\mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(It\mathcal{R}\text{-}\mathcal{SI}))$ is uf-cma assuming factoring is hard, but this was known [DKXY03]. Whether $It\mathcal{R}$-$\mathcal{SI}$ is imp-ca secure, and hence whether $It\mathcal{R}$-$\mathcal{IBI}$ is imp-ca secure, remains open.

In applying the $\mathsf{cSI\text{-}2\text{-}IBI}$ transform to the scheme, one must implement the random oracle $\mathrm{H} : \{0,1\}^* \to QR_N$ with care (here $QR_N = \{x^2 \bmod N \mid x \in \mathbb{Z}_N^*\}$ denotes the set of quadratic residues modulo N). Sampling random-looking elements from QR_N using standard hash functions may be hard without revealing a square root during the computation, since deciding whether an element $x \in \mathbb{Z}_N^*$ is a quadratic residue module N is assumed to be hard when the factorization of N is unknown.

This is not a problem in the abstract random oracle model, where one can simply mandate that H be chosen with domain $\{0,1\}^*$ and range QR_N, but the resulting scheme is difficult to instantiate. In practice, one would like to build H out of a cryptographic hash function like SHA-1 that has range $\{0,1\}^{160}$. Given N, there are standard techniques that yield a hash function with range $\mathbb{Z}_N^*$ [BR93a]. This is possible because membership in $\mathbb{Z}_N^*$ is decidable in polynomial time given N, and also $\mathbb{Z}_N^*$ is a "dense" subset of $\{0,1\}^k$ where k is the bit-length of N. However, there is no known way to build a function, computable in polynomial time given the input and N alone, that has range QR_N, because membership in the latter is not (known to be) decidable in polynomial time given N alone.

This problem can be overcome by using a Blum integer as the modulus (i.e. take $N = pq$ with $p \equiv q \equiv 3 \bmod 4$). Then it is well-known that -1 is a non-square modulo both p and q, and hence is a non-square modulo N

Algorithm $\mathsf{Kg}(1^k)$
$(N, p, q) \xleftarrow{R} \mathsf{K}_{\text{fact}}(1^k)$
Choose $\tau \geq \eta(p,q) - 1$; $g \xleftarrow{R} HQR_N$
$x_1 \xleftarrow{R} \mathbb{Z}_{2^m}$; $x_2 \xleftarrow{R} \mathbb{Z}_N^*$; $X \leftarrow g^{x_1} x_2^{2^{\tau+m}} \bmod N$
$pk \leftarrow ((N, \tau, g), X)$; $sk \leftarrow ((N, \tau, g), (x_1, x_2))$
Return (pk, sk)

Prover P		Verifier V
$y_1 \xleftarrow{R} \mathbb{Z}_{2^{\tau+m}}$; $y_2 \xleftarrow{R} \mathbb{Z}_N^*$		
$Y \leftarrow g^{y_1} y_2^{2^{\tau+m}} \bmod N$	$\xrightarrow{Y}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \{0,1\}^m$
$z_1 \leftarrow y_1 + cx_1 \bmod 2^{\tau+m}$		
$r \leftarrow \lfloor (y_1 + cx_1)/2^{\tau+m} \rfloor$		
$z_2 \leftarrow g^r y_2 x_2^c \bmod N$	$\xrightarrow{z_1, z_2}$	
		If $Y, z_2 \in \mathbb{Z}_N^*$ and $z_1 \in \mathbb{Z}_{2^{\tau+m}}$ and $g^{z_1} z_2^{2^{\tau+m}} \equiv Y X^c \bmod N$ then `acc` else `rej`

Figure 3.8: **The $\mathcal{FF}$-$\mathcal{SI}$ scheme.** The scheme is parameterized with modulus generator K_{fact} and an exponent $m \geq 1$. The prover P and verifier V are initialized with states $sk = ((N, \tau, g), (x_1, x_2))$ and $pk = ((N, \tau, g), X)$, respectively. The function $\eta(p, q)$ returns the smallest integer such that $2^{\eta(p,q)}$ divides $p - 1$ or $q - 1$, and the set $HQR_N = \{x^{2^{\eta(p,q)}} \bmod N \mid x \in \mathbb{Z}_N^*\}$.

with Jacobi symbol +1. As a consequence, for every element $x \in \mathbb{Z}_N^*[+1]$ (the elements of $\mathbb{Z}_N^*$ with Jacobi symbol +1), either x or $-x$ is a square modulo N. If we change the $It\mathcal{R}$-$\mathcal{SI}$ scheme such that $X_i \xleftarrow{R} \pm x_i^{2^m} \bmod N$ in the key generation and change the verification equation accordingly, then the random oracle maps bit strings to random elements of $\mathbb{Z}_N^*[+1]$ (the elements of $\mathbb{Z}_N^*$ with Jacobi symbol +1), which can be done using standard techniques. These changes do not affect security since the adversary gets even less information from the public key, and since the relaxation of the verification will at most double its impersonation advantage.

The Fischlin-Fischlin scheme. The $\mathcal{FF}$-$\mathcal{SI}$ scheme was introduced by Fischlin and Fischlin [FF02] as a fix to an attack they found on a scheme by Okamoto [Oka93]. In the key-generation algorithm of Figure 3.8, $\eta(p)$ denotes the largest integer such that $2^{\eta(p)}$ divides $p - 1$ and $\eta(p, q) = \max(\eta(p), \eta(q))$. $\mathcal{FF}$-$\mathcal{SI}$ is shown to be imp-pa, imp-aa, and imp-ca secure assuming factoring is hard [FF02]. The authors defined no IBI or IBS schemes. We can show that

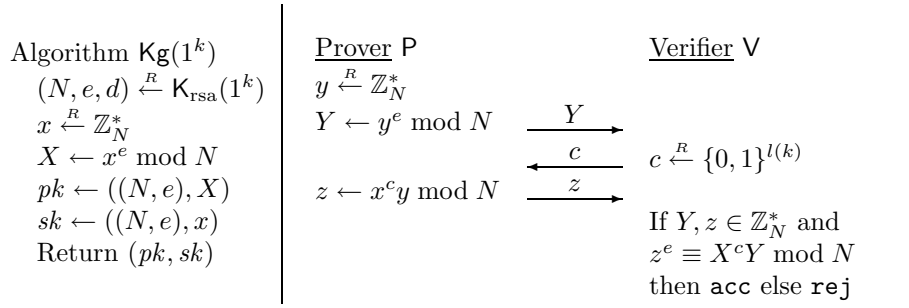


Figure 3.9: **The $\mathcal{GQ}$-$\mathcal{SI}$ scheme.** The scheme is parameterized with a prime-exponent RSA key generator $\mathsf{K}_{\mathrm{rsa}}$ and a superlogarithmic challenge length $l : \mathbb{N} \to \mathbb{N}$ such that $2^{l(k)} < e$ for all e output by $\mathsf{K}_{\mathrm{rsa}}$ on input 1^k. The prover P and verifier V are initialized with states $sk = ((N, e), x)$ and $pk = ((N, e), X)$, respectively.

$\mathcal{FF}$-$\mathcal{SI}$ is convertible, and we thus obtain $\mathcal{FF}$-$\mathcal{IBI}$ = $\mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{FF}\text{-}\mathcal{SI})$ and $\mathcal{FF}$-$\mathcal{IBS}$ = $\mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{FF}\text{-}\mathcal{SI}))$, and these are secure if factoring moduli generated by $\mathsf{K}_{\mathrm{fact}}$ is hard.

Let $HQR_N = \{x^{2^{\eta(p,q)}} \bmod N \mid x \in \mathbb{Z}_N^*\}$ denote the set of higher quadratic residues modulo N, which is also the subset of elements of $\mathbb{Z}_N^*$ of odd order. To show convertibility of $\mathcal{FF}$-$\mathcal{SI}$ we consider the relation $\mathbf{R} \subseteq (\mathbb{Z}_{2^m} \times \mathbb{Z}_N^*) \times HQR_N$ described by (N, g, τ) and containing tuples $((x_1, x_2), X)$ such that $g^{x_1} x_2^{2^{\tau+m}} \equiv X \bmod N$. The trapdoor is the factorization of N. Regularity holds since each higher quadratic residue has exactly $2^{\eta(p)+\eta(q)}$ different $2^{\eta(p,q)}$-th roots mod N, and hence also (because squaring is a permutation over HQR_N and $\tau + m \geq \eta(p, q)$) $2^{\eta(p)+\eta(q)}$ different $2^{\tau+m}$-th roots modulo N. Pair sampling involves choosing x_1, x_2 at random and computing $X = g^{x_1} x_2^{2^{\tau+m}}$.

Similar to $It\mathcal{R}$-$\mathcal{SI}$, the application of $\mathsf{cSI\text{-}2\text{-}IBI}$ to this scheme must be done with care when implementing the random oracle $H : \{0,1\}^* \to HQR_N$. Again we can require N to be a Blum integer, compute X as $\pm g^{x_1} x_2^{2^{\tau+m}} \bmod N$ in the Kg algorithm, and relax the verification to accept whenever $g^{z_1} z_2^{2^{\tau+m}} \equiv \pm Y X^c \bmod N$, such that the new range of the random oracle becomes $\mathbb{Z}_N^*[+1]$.

3.5.2 Schemes based on RSA

All schemes based on RSA are described in terms of an RSA key generator $\mathsf{K}_{\mathrm{rsa}}$ as defined in Section 2.3.1. A *prime-exponent* key generator only outputs keys with e prime. Security of schemes is based on the hardness of the associated RSA problem, or the associated one-more RSA problem.

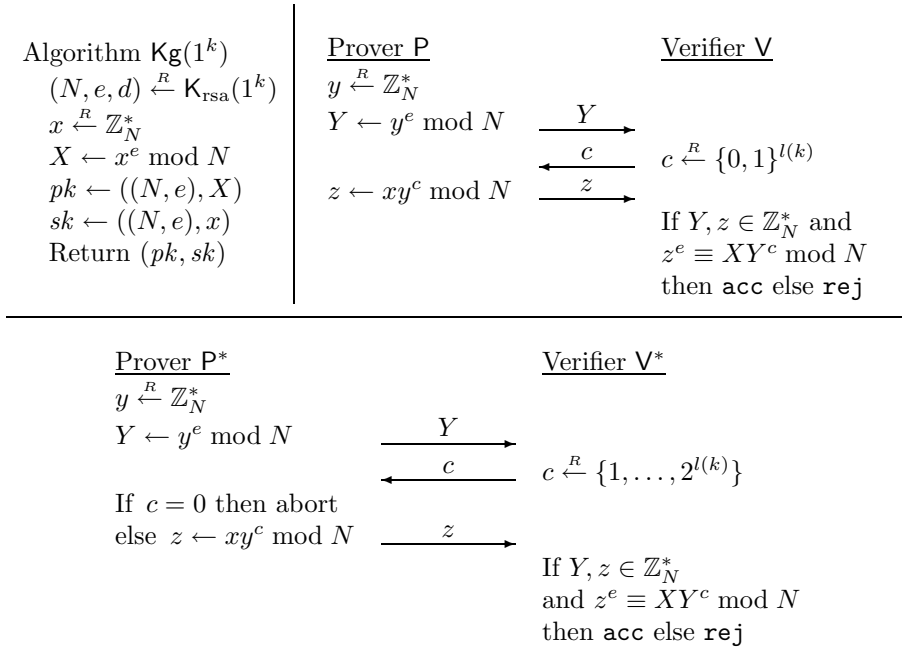


Figure 3.10: **The $\mathcal{SH}$-$\mathcal{SI}$ $= (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ and $\mathcal{SH}^*$-$\mathcal{SI}$ $= (\mathsf{Kg}, \mathsf{P}^*, \mathsf{V}^*)$ schemes.** Both schemes are specified in terms of a prime-exponent RSA key generator $\mathsf{K}_{\mathrm{rsa}}$ and a superlogarithmic challenge length $l : \mathbb{N} \to \mathbb{N}$ such that $2^{l(k)} < e$ for all e output by $\mathsf{K}_{\mathrm{rsa}}$ on input 1^k. The prover P and verifier V are initialized with states $sk = ((N, e), x)$ and $pk = ((N, e), X)$, respectively.

The Guillou-Quisquater scheme. The $\mathcal{GQ}$-$\mathcal{SI}$ scheme defined via Figure 3.9 is the standard one considered in the literature. Convertibility is easily seen by considering the relation $\mathbf{R} = \{(x, X) \mid x^e \equiv X \bmod N\}$, relation description $\langle \mathbf{R} \rangle = (N, e)$, and trapdoor d. Pair sampling involves choosing $x \xleftarrow{R} Z_N^*$ and computing $X \leftarrow x^e \bmod N$. We note that $\mathcal{GQ}$-$\mathcal{IBI}$ $= \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{GQ}\text{-}\mathcal{SI})$ is exactly the IBI scheme as presented by Guillou and Quisquater [GQ89], and $\mathcal{GQ}$-$\mathcal{IBS}$ $= \mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{GQ}\text{-}\mathcal{SI}))$ is exactly their IBS scheme. We know that $\mathcal{GQ}$-$\mathcal{SI}$ is imp-pa secure assuming RSA is one-way, and imp-aa and imp-ca secure assuming hardness of the one-more RSA problem [BP02]. Theorem 3.8 says that these results extend to $\mathcal{GQ}$-$\mathcal{IBI}$. (Also Corollary 3.11 says that $\mathcal{GQ}$-$\mathcal{IBS}$ is uf-cma assuming RSA is one-way, but this was known [DKXY03].)

The $\mathcal{SH}$ and $\mathcal{SH}^*$ schemes. Shamir [Sha84] introduced the concept of identity-based cryptography and presented the first IBS scheme, but did not define associated SI or IBI schemes. He also gave no security proof for his IBS scheme,

and none has been provided until now.

We surface the SI scheme $\mathcal{Sh}$-$\mathcal{SI}$ defined via Figure 3.10. One can check that $\mathcal{Sh}$-$\mathcal{IBS}$ = cSS-2-IBS(fs-I-2-S($\mathcal{Sh}$-$\mathcal{SI}$)) is exactly Shamir's IBS scheme [Sha84]. $\mathcal{Sh}$-$\mathcal{SI}$ is interesting both historically and technically. It turns out to be a "mirror-image" of $\mathcal{GQ}$-$\mathcal{SI}$ that closely resembles the latter. Convertibility of $\mathcal{Sh}$-$\mathcal{SI}$ follows from the convertibility of $\mathcal{GQ}$-$\mathcal{SI}$ since the two schemes have the same key-generation algorithm. Considering security, the first question to ask is whether $\mathcal{Sh}$-$\mathcal{SI}$ is honest-verifier zero-knowledge (HVZK). While this was obvious for $\mathcal{GQ}$-$\mathcal{SI}$ (as in fact it usually is, if true for an SI scheme), it is not apparent at first glance for $\mathcal{Sh}$-$\mathcal{SI}$, and one might suspect that the scheme is not HVZK. However, using a trick involving greatest common divisors and the extended Euclidean algorithm, we show that $\mathcal{Sh}$-$\mathcal{SI}$ is statistical (not perfect) HVZK. We also show that it is a proof of knowledge and thereby obtain the following:

Theorem 3.16 *The $\mathcal{Sh}$-$\mathcal{SI}$ is* imp-pa *secure assuming one-wayness of the underlying RSA key generator* $\mathsf{K}_{\mathrm{rsa}}$. ∎

Proof: The $\mathcal{Sh}$-$\mathcal{SI}$ scheme is statistical honest-verifier zero-knowledge since the following algorithm simulates communication transcripts using only the public key:

Algorithm Conv-sim$((N, e), X)$
  $c \xleftarrow{R} \{0,1\}^{l(k)}$
  Compute $a, b \in \mathbb{Z}$ such that $ac + be = 1$ (via extended Euclidean alg.)
  $y \xleftarrow{R} Z_N^*$; $Y \leftarrow X^{-a} \cdot y^e \bmod N$; $z \leftarrow X^b \cdot y^c \bmod N$
  Return (Y, c, z).

The transcripts generated by Conv-sim are correctly distributed since Y is uniformly distributed over $\mathbb{Z}_N^*$, c is uniformly distributed over $\{0,1\}^{l(k)}$ and z is the unique element of $\mathbb{Z}_N^*$ such that $z^e \equiv XY^c \bmod N$ because $z^e \equiv X^{be}y^{ec} \equiv X^{ac+be}Y^c \bmod N$. The second line of the algorithm may fail if $\gcd(c, e) \neq 1$. However, since e is prime with $2^{l(k)} < e$, the only problematic value is $c = 0$, which occurs only with negligible probability $2^{-l(k)}$ when the challenge length l is super-logarithmic in the security parameter.

The protocol is also a proof of knowledge of x, because from two valid challenge-response pairs $(c_1, z_1), (c_2, z_2)$ for the same commitment Y, one can extract the secret key x as follows. Use the extended Euclidean algorithm to compute $a, b \in Z$ such that $a(c_1 - c_2) + be = 1$. Since $(z_1/z_2)^e \equiv Y^{c_1 - c_2} \bmod N$, it holds that $Y \equiv Y^{a(c_1-c_2)+be} \equiv ((z_1/z_2)^a Y^b)^e \bmod N$, so that we can let $y \leftarrow (z_1/z_2)^a Y^b \bmod N$ and compute x as $z_1 y^{-c_1} \bmod N$. The extraction does not work if $\gcd(c_1 - c_2, e) > 1$, but since e is prime, this only occurs when $c_1 = c_2$, which again happens with negligible probability for super-logarithmic challenge length. ∎

Corollary 3.11 now implies that $\mathcal{SH}$-$I\mathcal{BS}$ is uf-cma secure under the same assumptions.

However, $\mathcal{SH}$-$\mathcal{SI}$ scheme is trivially insecure under active attacks, since the cheating verifier can learn the secret key by sending a zero challenge. But this minor weakness is easily fixed by "removing" the zero challenge. We define in Figure 3.10 a modified scheme we denote $\mathcal{SH}^*$-$\mathcal{SI}$. This scheme turns out to have security attributes analogous to those of $\mathcal{GQ}$-$\mathcal{SI}$ in that we can show the following:

Theorem 3.17 *The $\mathcal{SH}^*$-$\mathcal{SI}$ scheme is* imp-pa *secure under the RSA assumption associated to the underlying RSA key generator* $\mathsf{K}_{\mathrm{rsa}}$*, and* imp-aa *and* imp-ca *secure under the one-more RSA assumption relative to* $\mathsf{K}_{\mathrm{rsa}}$*.* ∎

Proof: The imp-pa security of the $\mathcal{SH}^*$-$\mathcal{SI}$ scheme under passive attack follows from the fact that it is perfect honest-verifier zero-knowledge and a proof of knowledge of x. Conversations can be simulated by an algorithm similar to $\mathsf{Conv\text{-}sim}$ in the proof of Theorem 3.16 but drawing c from $\{1, \ldots, 2^{l(k)}\}$. Extracting x is done exactly as in the proof of Theorem 3.16.

As one might expect, the proof under active and concurrent attack is very similar to the proof of the GQ identification scheme [BP02]. Given imp-ca adversary $\mathsf{A} = (\mathsf{CV}, \mathsf{CP})$ for the $\mathcal{SH}^*$-$\mathcal{SI}$ scheme, we construct a one-more RSA adversary B as follows. On input (N, e), B queries its challenge oracle the first time and stores the output as X. It then runs CV on input $pk = ((N, e), X)$. When CV requests to interact with a new prover session s, B queries its challenge oracle for a fresh target point Y_s and returns Y_s to CV. When confronted with challenge $c_s \neq 0$, B uses the inversion oracle to compute $z_s \leftarrow \textsc{Inv}(XY_s^{c_s} \bmod N)$ and returns it to CV. At the end of its execution, CV outputs initial state St_{CP} for the cheating prover CP.

Algorithm B then runs CP in a reset experiment as in Lemma 3.15 to generate two communication transcripts $(Y, \tilde{c}_1, \tilde{z}_1)$ and $(Y, \tilde{c}_2, \tilde{z}_2)$ where the challenges $\tilde{c}_1, \tilde{c}_2$ are uniformly distributed over S_1. With probability $\Pr[res(St_{\mathsf{CP}}, pk) = 1]$ these will both be accepting transcripts and $\tilde{c}_1 \neq \tilde{c}_2$. Moreover, since e is prime and $2^{l(k)} < e$, we can compute $a, b \in \mathbb{Z}$ such that $a(\tilde{c}_1 - \tilde{c}_2) + be = 1$ and compute $x \in \mathbb{Z}_N^*$ such that $x^e \equiv X \bmod N$ as in the proof of Theorem 3.16. Inversions of all other target points Y_s are either computed using the inversion oracle for unfinished sessions s, or are computed by applying the gcd trick again to get a, b such that $ac_s + be = 1$ and using the fact that $y_s \equiv y_s^{ac_s+be} \equiv (z_s/x)^a Y^b \bmod N$.

In summary, B needed one target point and one inversion query for each prover session, but succeeded in inverting X without the help of the inversion oracle, so it wins the game whenever the rewinding experiment succeeded. Using the

Reset Lemma, we have

$$\begin{aligned}\mathbf{Adv}^{\text{imp-}ca}_{\mathcal{SH}^*\text{-}\mathcal{SI},\mathsf{A}}(k) &= acc(St_{\mathsf{CP}}, pk) \\ &\leq 2^{-l(k)} + \sqrt{res(St_{\mathsf{CP}}, pk)} \\ &\leq 2^{-l(k)} + \sqrt{\mathbf{Adv}^{\text{1m-rsa}}_{\mathsf{K}_{\text{rsa}},\mathsf{B}}(k)}. \qquad (3.7)\end{aligned}$$

∎

We obtain the usual consequences for the security of $\mathcal{SH}^*\text{-}\mathcal{IBI} = \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{SH}^*\text{-}\mathcal{SI})$ and $\mathcal{SH}^*\text{-}\mathcal{IBS} = \mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{SH}^*\text{-}\mathcal{SI}))$.

The $\mathcal{OKRSA}$ scheme. Okamoto [Oka93] presented an RSA-based SI scheme and a related RSA-based IBI scheme. He proved the former imp-pa and imp-aa secure assuming factoring is hard, and the proofs extend to establish imp-ca as well. However, he did not prove the IBI scheme secure, a gap we fill.

The $\mathcal{OKRSA}\text{-}\mathcal{SI}$ scheme defined in Figure 3.11 is the SI scheme mentioned above. We observe that $\mathcal{OKRSA}\text{-}\mathcal{IBI} = \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{OKRSA}\text{-}\mathcal{SI})$ is exactly Okamoto's RSA-based IBI scheme [Oka93]. To show the security of the $\mathcal{OKRSA}\text{-}\mathcal{IBI}$ and $\mathcal{OKRSA}\text{-}\mathcal{IBS} = \mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{OKRSA}\text{-}\mathcal{SI}))$ schemes, it suffices to show that $\mathcal{OKRSA}\text{-}\mathcal{SI}$ is convertible. For this, the relation has description $\langle \mathbf{R} \rangle = (N, e, g)$, and contains tuples $((x_1, x_2), X) \in (\mathbb{Z}_e \times \mathbb{Z}_N^*) \times \mathbb{Z}_N^*$ such that $X \equiv g^{x_1} x_2^e \bmod N$. The trapdoor is d such that $ed \equiv 1 \bmod \varphi(N)$. Pair sampling involves choosing x_1, x_2 at random and computing $X \equiv g^{x_1} x_2^e$.

The $\mathcal{Gir}$ scheme. Girault [Gir90] proposed an SI scheme that we have defined in Figure 3.12 and named $\mathcal{Gir}\text{-}\mathcal{SI}$. He also proposed a related IBI scheme. (These schemes are inspired by the Schnorr identification scheme [Sch90] but use a modulus $N = pq$ where p, q are of the special form $p = 2fp' + 1$ and $q = 2fq' + 1$ such that f, p', q', p, q are all primes.) This IBI scheme did not use hash functions, which lead to an attack and later a fix [SSN98]. The fixed IBI scheme turns out to be exactly $\mathcal{Gir}\text{-}\mathcal{IBI} = \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{Gir}\text{-}\mathcal{SI})$.

$\mathcal{Gir}\text{-}\mathcal{SI}$ is convertible with relation $\mathbf{R} = \{((P, s), X) \mid P^e \equiv X^{-1} h^{-s} \bmod N\}$ described by (N, e, h, f). The trapdoor is $d \equiv e^{-1} \bmod \varphi(N)$. Pair sampling involves choosing P and s at random and computing X as $P^{-e} h^{-s} \bmod N$. However, this does not help here because we found that all schemes in the family are insecure. In particular, $\mathcal{Gir}\text{-}\mathcal{SI}$ is not even imp-pa secure, and neither is the fixed IBI scheme $\mathcal{Gir}\text{-}\mathcal{IBI}$. The identity-based signature scheme $\mathcal{Gir}\text{-}\mathcal{IBS} = \mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{Gir}\text{-}\mathcal{IBI}))$ is not uf-cma secure either.

Theorem 3.18 (Insecurity of the $\mathcal{Gir}$ Family) *The $\mathcal{Gir}\text{-}\mathcal{SI}$ scheme depicted in Figure 3.12 and the $\mathcal{Gir}\text{-}\mathcal{IBI} = \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{Gir}\text{-}\mathcal{SI})$ scheme [Gir90, SSN98] are insecure against impersonation under passive, active and concurrent attack. The $\mathcal{Gir}\text{-}\mathcal{SS} = \mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{Gir}\text{-}\mathcal{SI})$ and the $\mathcal{Gir}\text{-}\mathcal{IBS} = \mathsf{cSS\text{-}2\text{-}IBS}(\mathcal{Gir}\text{-}\mathcal{SS})$ schemes are universally forgeable under known-message attack.* ∎

Algorithm $\mathsf{Kg}(1^k)$
$(N, e, d) \xleftarrow{R} \mathsf{K}_{\mathrm{rsa}}(1^k)$; $g \xleftarrow{R} \mathbb{Z}_N^*$
$x_1 \xleftarrow{R} \mathbb{Z}_e$; $x_2 \xleftarrow{R} \mathbb{Z}_N^*$; $X \leftarrow g^{-x_1} x_2^{-e} \bmod N$
$pk \leftarrow ((N, e, g), X)$; $sk \leftarrow ((N, e, g), (x_1, x_2))$
Return (pk, sk)

Prover P		Verifier V
$y_1 \xleftarrow{R} \mathbb{Z}_e$; $y_2 \xleftarrow{R} \mathbb{Z}_N^*$		
$Y \leftarrow g^{y_1} y_2^e \bmod N$	$\xrightarrow{Y}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \{0,1\}^{l(k)}$
$z_1 \leftarrow y_1 + c x_1 \bmod e$		
$r \leftarrow \lfloor (y_1 + c x_1)/e \rfloor$		
$z_2 \leftarrow g^r y_2 x_2^c \bmod N$	$\xrightarrow{z_1, z_2}$	
		If $Y, z_2 \in \mathbb{Z}_N^*$ and $z_1 \in \mathbb{Z}_e$ and $Y \equiv g^{z_1} z_2^e X^c \bmod N$ then `acc` else `rej`

Figure 3.11: **The $\mathcal{OKRSA}$-$\mathcal{SI}$ scheme.** The scheme is parameterized with a prime-exponent RSA generator $\mathsf{K}_{\mathrm{rsa}}$ and a challenge length $l : \mathbb{N} \to \mathbb{N}$ such that $2^{l(k)} < e$ for any e output by $\mathsf{K}_{\mathrm{rsa}}(1^k)$. The prover P and verifier V are initialized with states $sk = ((N, e, g), (x_1, x_2))$ and $pk = ((N, e, g), X)$, respectively.

Proof: We attack only the $\mathcal{Gir}$-$\mathcal{IBS}$ scheme, since the insecurity of the SI, IBI, and SS schemes then follows as a consequence. In the $\mathcal{Gir}$-$\mathcal{IBS}$ scheme, a signature of a user I on a message M under the master public key $mpk = (N, e, h, f)$ is a tuple (P, Y, z) such that $Y \equiv h^z (P^e \cdot \mathrm{H}_1(I))^{\mathrm{H}_2(P \| Y \| M)} \bmod N$, where H_1 is the random oracle associated to the $\mathsf{cSI\text{-}2\text{-}IBI}$ transform and H_2 is the random oracle associated to the $\mathsf{fs\text{-}I\text{-}2\text{-}S}$ transform. The flaw at the heart of the attack is that in the subgroup generated by g, computing RSA inverses is easy because the order f of the subgroup is known. Given a valid signature (P_1, Y_1, z_1) for message M_1 and identity I, an adversary can forge I's signature for any message M_2 as follows. It first computes $d' \leftarrow e^{-1} \bmod f$ and $g' \leftarrow h^{d'} \bmod N$. Because h is of order f, we have $g \equiv g' \bmod N$. It also computes $S' \leftarrow (P_1^e \cdot \mathrm{H}_1(I))^{d'} \bmod N$ such that

$$\begin{aligned} S' &\equiv \left(\mathrm{H}_1(I)^{-1} S^e \cdot \mathrm{H}_1(I)\right)^{d'} \bmod N \\ &\equiv S \bmod N. \end{aligned}$$

Then, it chooses s_2 from $\mathbb{Z}_f$ and computes $P_2 \leftarrow P_1 {S'}^{-1} {g'}^{-s_2} \bmod N$. Since $P_2 \equiv \mathrm{H}_1(I)^{-d} g^{-s_2} \bmod N$, the pair (P_2, s_2) might have been output by the UKg

Algorithm $\mathsf{Kg}(1^k)$
$(N, e, d, f) \xleftarrow{R} \mathsf{K}_{\mathrm{rsa}}(1^k)$
Choose $g \in \mathbb{Z}_N^*$ of order f ; $h \leftarrow g^e \bmod N$
$s \xleftarrow{R} \mathbb{Z}_f$; $S \leftarrow g^{-s} \bmod N$
$X \xleftarrow{R} \mathbb{Z}_N^*$; $P \leftarrow X^{-d} S \bmod N$
$pk \leftarrow ((N, e, h, f), X)$; $sk \leftarrow ((N, e, h, f), (P, s))$
Return (pk, sk)

Prover P		Verifier V
$y \xleftarrow{R} \mathbb{Z}_f$		
$Y \leftarrow h^y \bmod N$	$\xrightarrow{P, Y}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \{0,1\}^{l(k)}$
$z \leftarrow y + sc \bmod f$	$\xrightarrow{z}$	
		If $P, Y \in \mathbb{Z}_N^*$ and $z \in \mathbb{Z}_f$ and $h^z (P^e X)^c \equiv Y \bmod N$ then `acc` else `rej`

Figure 3.12: **The *Gir-SI* scheme.** The scheme is parameterized with a challenge length $l(k)$ and an RSA key generator $\mathsf{K}_{\mathrm{rsa}}$ that returns the modulus $N = pq$ with p, q of the form $p = 2fp' + 1$ and $q = 2fq' + 1$ where f, p', q', p, q are all primes. The generator outputs N, e, d as well as f. The prover P and verifier V are initialized with states $sk = ((N, e, h, f), (P, s))$ and $pk = ((N, e, h, f), X)$, respectively.

algorithm as part of the user secret key corresponding to identity I. Therefore, any signature the adversary generates using this pair will be considered valid for identity I. The adversary now follows the normal signing algorithm to compute the forgery: it chooses y_2 from $\mathbb{Z}_q$, sets $Y_2 \leftarrow h^{y_2} \bmod N$, computes $z_2 \leftarrow y_2 + s_2 \mathrm{H}_2(P_2 \| Y_2 \| M_2) \bmod f$. The forgery is (P_2, Y_2, z_2). ∎

It is natural to consider counteracting the above attack by removing f from the public key. While this might work for the SI scheme, it does not for the IBI (or IBS) scheme. The reason is that, since f still has to be included in each user's secret key, an adversary can easily extract it by corrupting one identity.

We stress that the scheme broken here is *not* the (perhaps better-known) SI scheme by Girault based on discrete logarithms [Gir91] that forms the basis of the GPS identification scheme [PS98, BBB+00] and was accepted as a NESSIE recommendation [NES03].

Algorithm $\mathsf{Kg}(1^k)$
$(\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}) \xleftarrow{R} \mathsf{K}_{\text{pair}}(1^k)$
$s \xleftarrow{R} \mathbb{Z}_q$; $S \leftarrow sP$; $U \xleftarrow{R} \mathbb{G}_1$; $V \leftarrow sU$
$pk \leftarrow ((\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}, S), U)$; $sk \leftarrow ((\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}, S), V)$
Return (pk, sk)

Prover P		Verifier V
$y \xleftarrow{R} \mathbb{Z}_q$; $Y \leftarrow yP$	$\xrightarrow{Y}$	
	$\xleftarrow{C}$	$C \xleftarrow{R} \mathbb{G}_1$
$Z \leftarrow yC + V$	$\xrightarrow{Z}$	
		If $Y, Z \in \mathbb{G}_1$ and $\hat{e}(Z, P) = \hat{e}(U, S)\hat{e}(C, Y)$ then `acc` else `rej`

Prover P		Verifier V
$y \xleftarrow{R} \mathbb{Z}_q$; $\alpha \leftarrow \hat{e}(P, P)^y$	$\xrightarrow{\alpha}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \mathbb{Z}_q$
$Z \leftarrow yP + cV$	$\xrightarrow{Z}$	
		If $\alpha \in \mathbb{G}_2$ and $Z \in \mathbb{G}_1$ and $\hat{e}(Z, P) = \alpha \cdot \hat{e}(U, S)^c$ then `acc` else `rej`

Prover P		Verifier V
$y \xleftarrow{R} \mathbb{Z}_q$; $Y \leftarrow yU$	$\xrightarrow{Y}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \mathbb{Z}_q$
$Z \leftarrow (y + c)V$	$\xrightarrow{Z}$	
		If $Y, Z \in \mathbb{G}_1$ and $\hat{e}(Z, P) = \hat{e}(Y + cU, S)$ then `acc` else `rej`

Figure 3.13: **SI schemes surfaced from pairing-based IBS schemes.** All schemes use the same key generation algorithm Kg. Presented here are (from top to bottom) the *SOK-SI*, *Hs-SI* and *ChCh-SI* schemes. The provers P and verifiers V are initialized with states $sk = ((\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}, S), V)$ and $pk = ((\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}, S), U)$, respectively.

3.5.3 Pairing-Based Schemes

Many recent papers propose pairing-based IBS schemes [SOK00, CC03, Yi03, Pat02, Hes03] (we found the schemes independently published by [CC03] and [Yi03] actually to be equivalent). Barring [CC03], none of these papers prove their scheme secure. (Some papers contained proofs in weak models however [Hes03, Yi03], another claimed but not did not state a proof [Pat02].) However, the scheme of Hess [Hes03] was proven secure by Dodis et al. [DKXY03].

None of these papers define SI or IBI schemes. We surface $\mathcal{SOK}$-$\mathcal{SI}$ [SOK00], $\mathcal{ChCh}$-$\mathcal{SI}$ [CC03, Yi03] and $\mathcal{Hs}$-$\mathcal{SI}$ [Hes03], as defined by Figure 3.13. The key generation algorithm is the same for all these schemes and uses a pairing generator K_{pair} as defined in Section 2.3. The $\mathcal{ChCh}$-$\mathcal{IBS}$ = cSS-2-IBS(fs-I-2-S($\mathcal{ChCh}$-$\mathcal{SI}$)) and $\mathcal{Hs}$-$\mathcal{IBS}$ = cSS-2-IBS(fs-I-2-S($\mathcal{Hs}$-$\mathcal{SI}$)) schemes are exactly the IBS schemes of the original papers, while $\mathcal{SOK}$-$\mathcal{IBS}$ = cSS-2-IBS(fs-I-2-S($\mathcal{SOK}$-$\mathcal{SI}$)) is slightly different from the scheme of Sakai et al. [SOK00]. Paterson's scheme [Pat02] does not seem to be related to any convertible SI scheme, leaving its security as an open problem.

We now show that all these pairing-based SI schemes are convertible. Since they have the same key-generation algorithm, a common argument applies. The relation is $\{(V,U) \in \mathbb{G}_1 \times \mathbb{G}_1 \,|\, \hat{e}(V,P) = \hat{e}(U,S)\}$, described by $\langle \mathbf{R} \rangle = (\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}, S)$. The trapdoor is s such that $S = sP$. Pair sampling is done by choosing $r \xleftarrow{R} \mathbb{Z}_q$ and computing the pair (rP, rS), and inversion can be done by raising elements of $\mathbb{G}_1$ to the power $s^{-1} \bmod q$.

Theorem 3.19 *$\mathcal{SOK}$-$\mathcal{SI}$ and $\mathcal{ChCh}$-$\mathcal{SI}$ are* imp-pa *secure assuming that the computational Diffie-Hellman problem in the group $\mathbb{G}_1$ associated to K_{pair} is hard.* ∎

Proof: We prove security against impersonation under passive attack by showing that all three schemes are honest-verifier zero-knowledge and proofs of knowledge for V. The former can be seen from the conversation simulators given in Figure 3.14. It is easily verified that their outputs are correctly distributed. We demonstrate the proof of knowledge property by showing how any cheating prover CP can be used to extract the prover's secret V. For the $\mathcal{SOK}$-$\mathcal{SI}$ scheme, the extractor chooses $c \xleftarrow{R} \mathbb{Z}_q$ upon receiving Y from CP, and sends $C \leftarrow cP$ as the challenge. From CP's response Z, the extractor computes V as $Z - cY$. The extractor of the two other schemes runs the cheating prover in a reset experiment to obtain two responses Z_1, Z_2 to randomly chosen challenges c_1, c_2 for the same commitment Y (or α). If both transcripts are valid, V can be computed as $(c_1 - c_2)^{-1}(Z_1 - Z_2)$. Using the Reset Lemma, we obtain the

Simulator for $\mathcal{SOK}$-$\mathcal{SI}$:	Simulator for $\mathcal{Hs}$-$\mathcal{SI}$:	Simulator for $\mathcal{ChCh}$-$\mathcal{SI}$:
$y \xleftarrow{R} \mathbb{Z}_q$; $Y \leftarrow yS$	$Z \xleftarrow{R} \mathbb{G}_1$	$z \xleftarrow{R} \mathbb{Z}_q$; $Z \leftarrow zS$
$z \xleftarrow{R} \mathbb{Z}_q$; $Z \leftarrow zS$	$c \xleftarrow{R} \mathbb{Z}_q$	$c \xleftarrow{R} \mathbb{Z}_q$
$C \leftarrow y^{-1}(zP - U)$	$\alpha \leftarrow \hat{e}(Z,P)\hat{e}(U,S)^{-c}$	$Y \leftarrow zP - cU$
Return (Y, C, Z)	Return (α, c, Z)	Return (Y, c, Z)

Figure 3.14: Conversation simulator algorithms for the pairing-based schemes.

following bounds on the advantage of any imp-pa adversary A:

$$\begin{aligned} \mathbf{Adv}^{\text{imp-pa}}_{\mathcal{SOK}\text{-}\mathcal{SI},\mathsf{A}}(k) &\leq \mathbf{Adv}^{\text{cdh}}_{\mathsf{K}_{\text{pair}},B}(k) \\ \mathbf{Adv}^{\text{imp-pa}}_{\mathcal{SI},\mathsf{A}}(k) &\leq 2^{-k+1} + \sqrt{\mathbf{Adv}^{\text{cdh}}_{\mathsf{K}_{\text{pair}},B}(k)} \quad \text{for } \mathcal{SI} \in \{\mathcal{Hs}\text{-}\mathcal{SI}, \mathcal{ChCh}\text{-}\mathcal{SI}\} \ . \end{aligned}$$

Corollary 3.11 implies that $\mathcal{ChCh}$-$\mathcal{IBS}$, $\mathcal{SOK}$-$\mathcal{IBS}$ and $\mathcal{Hs}$-$\mathcal{IBS}$ are uf-cma secure IBS schemes, but of these only the result about $\mathcal{SOK}$-$\mathcal{IBS}$ is new. However, we are also able to prove the following:

Theorem 3.20 *$\mathcal{ChCh}$-$\mathcal{SI}$ and $\mathcal{Hs}$-$\mathcal{SI}$ are* imp-aa *and* imp-ca *secure assuming that the one-more computational Diffie-Hellman problem in the group $\mathbb{G}_1$ associated to K_{pair} is hard.*

Proof: The way to construct a one-more CDH algorithm B out of an imp-aa/ca adversary $\mathsf{A} = (\mathsf{CV}, \mathsf{CP})$ is actually very similar for the $\mathcal{ChCh}$-$\mathcal{SI}$ and $\mathcal{Hs}$-$\mathcal{SI}$ schemes. We present a single construction here and mention the differences as they occur. When run on input (P, aP), algorithm B assigns $S \leftarrow aP$, queries the challenge oracle a first time to get $U \leftarrow \textsc{Chall}(\varepsilon)$, and runs CV on input $pk = ((\mathbb{G}_1, \mathbb{G}_2, P, q, \hat{e}, S), U)$. Each time CV asks for an interaction with a new prover session i, it queries the challenge oracle to get $Y_i \leftarrow \textsc{Chall}(\varepsilon)$. This value is returned to the cheating verifier for the $\mathcal{ChCh}$-$\mathcal{SI}$ scheme, while $\alpha_i \leftarrow \hat{e}(Y_i, S)$ is returned for the $\mathcal{Hs}$-$\mathcal{SI}$ scheme. Upon receiving the challenge c_i from CV, the one-more CDH adversary B uses its CDH oracle to compute $Z_i \leftarrow \textsc{Cdh}(Y_i + c_iU)$ and returns it to CV. The validity of this response can be verified by observing that for the $\mathcal{ChCh}$-$\mathcal{SI}$ scheme it holds that $\hat{e}(Z_i, P) = \hat{e}(a(Y_i + c_iU), P) = \hat{e}(Y_i + c_iU, S)$, and for the $\mathcal{Hs}$-$\mathcal{SI}$ scheme that $\hat{e}(Z_i, P) = \hat{e}(a(Y_i + c_iU), P) = \hat{e}(Y_i, S)\hat{e}(c_iU, S) = \alpha_i \cdot \hat{e}(U, S)^{c_i}$. When CV outputs the initial state St_{CP} for the cheating prover, B extracts a value V from CP such that $V = aU$ by running CP in a reset experiment as in the proof of Theorem 3.19. This is the solution to B's first challenge, and it can compute solutions to all other challenges as $Q_i \leftarrow Z_i - c_iV$. (The solution for Y_i in unfinished prover sessions

can be queried directly from the CDH oracle.) In summary, if CV interacted with n different prover sessions, then B succeeded in solving $n+1$ challenges using only n CDH queries, and hence wins the game. Therefore, by Lemma 3.15, the advantage of an imp-aa/ca A for $\mathcal{SI} \in \{\mathcal{ChCh}\text{-}\mathcal{SI}, \mathcal{Hs}\text{-}\mathcal{SI}\}$ is bounded by

$$\mathbf{Adv}^{\text{imp-aa/ca}}_{\mathcal{SI},\mathsf{A}}(k) \leq 2^{-k+1} + \sqrt{\mathbf{Adv}^{\text{1m-cdh}}_{\mathsf{K}_{\text{pair}},B}(k)} \; .$$

∎

Theorem 3.8 implies that the $\mathcal{ChCh}\text{-}\mathcal{IBI}$ and $\mathcal{Hs}\text{-}\mathcal{IBI}$ schemes are imp-aa and imp-ca secure assuming that the one-more CDH problem in the group $\mathbb{G}_1$ associated to K_{pair} is hard. Thus, we obtain new, pairing-based IBI schemes with proofs of security.

$\mathcal{SOK}\text{-}\mathcal{SI}$ and $\mathcal{SOK}\text{-}\mathcal{IBI}$ are insecure under active and concurrent attacks: upon receiving a commitment Y, an adversary can choose $c \xleftarrow{R} \mathbb{Z}_q$, submit $C \leftarrow cP$ as the challenge, and compute the prover's secret key from the response Z as $V \leftarrow Z - cY$. As indicated above, $\mathcal{SOK}\text{-}\mathcal{IBS}$, that we prove secure, is slightly different from the published IBS scheme [SOK00]. It is unclear whether the latter can be proved secure, so $\mathcal{SOK}\text{-}\mathcal{IBS}$ might be preferable to the original one. This highlights a benefit of our framework, namely that we can obtain provable schemes in a systematic way.

3.5.4 A Scheme based on Discrete Logarithms

THE $\mathcal{Beth}^t$ SCHEME. The $\mathcal{Beth}^t\text{-}\mathcal{SI}$ scheme defined in Figure 3.15 was surfaced from an IBI scheme by Beth [Bet88]. It is parameterized with a discrete logarithm group generator K_{dlog} as defined in Section 2.3.2. The $\mathcal{Beth}^t\text{-}\mathcal{IBI} = \mathsf{cSI\text{-}2\text{-}IBI}(\mathcal{Beth}^t\text{-}\mathcal{SI})$ scheme is a more efficient version of the IBI scheme actually presented [Bet88]. In these schemes, the prover proves knowledge of an ElGamal signature [El 84] of his identity. Beth [Bet88] gives no security proofs, but here we obtain one for the special case of $\mathcal{Beth}^1\text{-}\mathcal{IBI}$.

It would be tempting to say that the $\mathcal{Beth}^t\text{-}\mathcal{SI}$ scheme is convertible with relation $\mathbf{R} = \{((R, s_1, \ldots, s_t), (h_1, \ldots, h_t)) \in (\mathbb{G} \times \mathbb{Z}_q^t) \times \mathbb{Z}_q^t \mid X_i^R R^{s_i} \equiv g^{h_i} \text{ for } i = 1 \ldots t\}$ described by $\langle \mathbf{R} \rangle = (\mathbb{G}, q, g, X_1, \ldots, X_t)$ and with trapdoor information $(x_1, \ldots, x_t)$ such that $g^{x_i} \equiv X_i$ for $i = 1 \ldots t$. In the case that $t = 1$ (and using x, X, h as a shorthand notation for x_1, X_1, h_1), pair sampling can be done by choosing a, b at random from $\mathbb{Z}_q$ and letting $R \leftarrow X^a g^b$, $s \leftarrow a^{-1} R \bmod q$ and $h \leftarrow bs \bmod q$. (This trick is actually related to the existential forgery attack on textbook-ElGamal signatures [El 84].) However, it is not clear how to sample the relation for $t > 1$, the problem being that the same R has to "fit" all X_i. Thus, while we know that $\mathcal{Beth}^1\text{-}\mathcal{SI}$ is a convertible SI scheme, we do not know whether the same holds true for $\mathcal{Beth}^t\text{-}\mathcal{SI}$ with $t > 1$.

We were unable to prove the $\mathcal{Beth}^1\text{-}\mathcal{SI}$ scheme under a "clean" mathemati-

Algorithm $\mathsf{Kg}(1^k)$
$(\mathbb{G}, q, g) \xleftarrow{R} \mathsf{K}_{\mathrm{dlog}}(1^k)$
$r \xleftarrow{R} \mathbb{Z}_q$; $R \leftarrow g^r$
For $i = 1 \ldots t$ do
$x_i \xleftarrow{R} \mathbb{Z}_q$; $X_i \leftarrow g^{x_i}$; $h_i \xleftarrow{R} Z_q$
Compute s_i such that $Rx_i + rs_i \equiv h_i \bmod q$
$pk \leftarrow ((\mathbb{G}, q, g, X_1, \ldots X_t), (h_1, \ldots h_t))$
$sk \leftarrow ((\mathbb{G}, q, g, X_1, \ldots X_t), (R, s_1, \ldots, s_t))$
Return (pk, sk)

Prover P		Verifier V
$y \xleftarrow{R} \mathbb{Z}_q$; $Y \leftarrow R^{-y}$	$\xrightarrow{R, Y}$	
	$\xleftarrow{c}$	$c = (c_1, \ldots, c_t) \xleftarrow{R} \left(\{0,1\}^{l(k)}\right)^t$
$z \leftarrow y + \sum_i c_i s_i \bmod q$	$\xrightarrow{z}$	
		If $R, Y \in \mathbb{G}$ and $z \in \mathbb{Z}_q$ and $g^{\sum_i c_i h_i} \equiv R^z Y \prod_i X_i^{c_i R}$ then `acc` else `rej`

Figure 3.15: **The $\mathcal{B}eth^t$-$\mathcal{SI}$ scheme.** The scheme is parameterized with the discrete-log group generator $\mathsf{K}_{\mathrm{dlog}}$, a key multiplicity $t \geq 1$ and and superlogarithmic challenge length $l : \mathbb{N} \to \mathbb{N}$ such that $2^{l(k)} < q$ for all q output by $\mathsf{K}_{\mathrm{dlog}}(1^k)$. The prover P and verifier V are run on initial states $sk = ((\mathbb{G}, q, g, X_1, \ldots, X_t), (R, s_1 \ldots s_t))$ and $pk = ((\mathbb{G}, q, g, X_1, \ldots X_t), (h_1 \ldots h_t))$, respectively.

cal assumption, but Theorem 3.21 proves the passive security of $\mathcal{B}eth^1$-$\mathcal{SI}$ under the assumption that the hashed-message ElGamal signature scheme, presented as $\mathcal{ElG}$-$\mathcal{SS}$ in Figure 3.16, is secure under the (very weak) notion of universal unforgeability under no-message attack in the random oracle model. The $\mathcal{ElG}$-$\mathcal{SS}$ scheme is a close variant of the Modified ElGamal signature scheme that was proven secure under the discrete logarithm assumption [PS00]. Bleichenbacher [Ble96] demonstrated an attack on the $\mathcal{ElG}$-$\mathcal{SS}$ scheme and showed how to counteract it by carefully choosing the public parameters or by restricting the values of valid signatures.

Theorem 3.21 *$\mathcal{B}eth^1$-$\mathcal{SI}$ is* imp-pa *secure assuming that the $\mathcal{ElG}$-$\mathcal{SS}$ scheme associated to $\mathsf{K}_{\mathrm{dlog}}$ is universally unforgeable under no-message attack in the random oracle model.* ∎

Proof: Given an imp-pa adversary $\mathsf{A} = (\mathsf{CV}, \mathsf{CP})$, we construct a universal forger F as follows. On input $pk = (\mathbb{G}, q, g, X)$, the forger first chooses $r \xleftarrow{R} \mathbb{Z}_q$, lets $R \leftarrow g^r$ and runs CV on input $((\mathbb{G}, q, g, X), \mathrm{H}(M))$. Note that since H

Algorithm $\mathsf{Kg}(1^k)$
$(\mathbb{G}, q, g) \xleftarrow{R} \mathsf{K}_{\mathrm{dlog}}(1^k)$
$x \xleftarrow{R} \mathbb{Z}_q$; $X \leftarrow g^x$
$pk \leftarrow (\mathbb{G}, q, g, X)$; $sk \leftarrow (\mathbb{G}, q, g, x)$
Return (pk, sk)

Algorithm $\mathsf{Sign}(sk, M : \mathrm{H})$
Parse sk as $(\mathbb{G}, q, g, x)$
$r \xleftarrow{R} \mathbb{Z}_q$; $R \leftarrow g^r$
Compute s such that
$Rx + rs \equiv \mathrm{H}(M) \bmod q$
Return (R, s)

Algorithm $\mathsf{Vf}(pk, M, \sigma : \mathrm{H})$
Parse pk as $(\mathbb{G}, q, g, X)$
Parse σ as (R, s)
If $X^R R^s \equiv g^{\mathrm{H}(M)}$
then return 1 else return 0

Figure 3.16: **The $\mathcal{ElG}$-$\mathcal{SS}$ scheme.** The scheme is parameterized with discrete logarithm group generator $\mathsf{K}_{\mathrm{dlog}}$. The signing and verification algorithms have access to a random oracle $\mathrm{H} : \{0,1\}^* \to \mathbb{Z}_q$.

is a random oracle, this public key is correctly distributed. It answers CV's conversation queries by each time choosing c and z at random from $\{0,1\}^{l(k)}$ and $\mathbb{Z}_q$, respectively, computing $Y \leftarrow g^{c\mathrm{H}(M)} R^{-z} X^{-cR}$ and returning (Y, c, z) as the transcript. When CV outputs St_{CP}, the forger runs the cheating prover in a reset experiment as in Lemma 3.15 to get commitment $\tilde{R}, Y$ and responses z_1, z_2 to challenges c_1, c_2 chosen at random from $\{0,1\}^{l(k)}$. Note that $\tilde{R}$ does not have to be equal to R used in the simulated transcripts. If the reset experiment is successful (meaning that both responses are valid and $c_1 \neq c_2$), the forger computes $s \leftarrow (c_1 - c_2)^{-1}(z_1 - z_2) \bmod q$ and outputs $\sigma = (\tilde{R}, s)$ as the forgery for M. By dividing the two verification equations of the reset experiment, it is easily seen that this is a valid signature for M. Due to the Reset Lemma, the imp-pa advantage of A is bounded by

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{Beth}^1\text{-}\mathcal{SI},\mathsf{A}}(k) \leq 2^{-l(k)} + \sqrt{\mathbf{Adv}^{\text{uuf-nma}}_{\mathcal{ElG}\text{-}\mathcal{SS},\mathsf{F}}(k)},$$

which is a negligible quantity for super-logarithmic functions $l(k)$, thereby concluding the proof. ∎

Theorem 3.8 implies that $\mathcal{Beth}^1$-$\mathcal{IBI}$ inherits the above security attributes, and Corollary 3.11 implies that $\mathcal{Beth}^1$-$\mathcal{IBS}$ = $\mathsf{cSS\text{-}2\text{-}IBS}(\mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{Beth}^1\text{-}\mathcal{SI}))$ is uf-cma secure under the same assumptions. The imp-aa and imp-ca security of $\mathcal{Beth}^1$-$\mathcal{SI}$ remains open, as does the security of $\mathcal{Beth}^t$-$\mathcal{SI}$ scheme with $t > 1$.

Algorithm $\mathsf{MKg}(1^k)$
$(\mathbb{G}, q, g_1) \xleftarrow{R} \mathsf{K}_{\mathrm{dlog}}(1^k)$
$\alpha \xleftarrow{R} \mathbb{Z}_q$; $g_2 \leftarrow g_1^{\alpha}$
$x_1, x_2 \xleftarrow{R} \mathbb{Z}_q$; $X \leftarrow g_1^{-x_1} g_2^{-x_2}$
$mpk \leftarrow (\mathbb{G}, q, g_1, g_2, X)$
$msk \leftarrow (\mathbb{G}, q, g_1, g_2, x_1, x_2)$
Return (mpk, msk)

Algorithm $\mathsf{UKg}(msk, I : \mathrm{H})$
$(\mathbb{G}, q, g_1, g_2, x_1, x_2) \leftarrow msk$
$r_1, r_2 \xleftarrow{R} \mathbb{Z}_q$; $R \leftarrow g_1^{r_1} g_2^{r_2}$
$h \leftarrow \mathrm{H}(R||I)$
$s_1 \leftarrow -r_1 - hx_1 \bmod q$
$s_2 \leftarrow -r_2 - hx_2 \bmod q$
$usk \leftarrow (\mathbb{G}, q, g_1, g_2, h, s_1, s_2)$
Return usk

Prover $\overline{\mathsf{P}}$		Verifier $\overline{\mathsf{V}}$
$S \leftarrow g_1^{-s_1} g_2^{-s_2}$		
$y_1, y_2 \xleftarrow{R} \mathbb{Z}_q$; $Y \leftarrow g_1^{y_1} g_2^{y_2}$	$\xrightarrow{h, S, Y}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \{0,1\}^{l(k)}$
$z_1 \leftarrow y_1 + cs_1 \bmod q$		
$z_2 \leftarrow y_2 + cs_2 \bmod q$	$\xrightarrow{z_1, z_2}$	
		If $h \in \{0,1\}^{l(k)}$ and $S, Y \in \mathbb{G}$ and $z_1, z_2 \in \mathbb{Z}_q$ and $Y \equiv g_1^{z_1} g_2^{z_2} S^c$ and $h = \mathrm{H}(X^h S||I)$ then `acc` else `rej`

Figure 3.17: **The *OKDL-IBI* scheme.** The scheme is parameterized by discrete-log group generator $\mathsf{K}_{\mathrm{dlog}}$ and superlogarithmic challenge length $l : \mathbb{N} \to \mathbb{N}$ such that $2^{l(k)} < q$ for all q output by $\mathsf{K}_{\mathrm{dlog}}(1^k)$. The prover $\overline{\mathsf{P}}$ and verifier $\overline{\mathsf{V}}$ are run on initial states $usk = (\mathbb{G}, q, g_1, g_2, h, s_1, s_2)$ and (mpk, I) where $mpk = (\mathbb{G}, q, g_1, g_2, X)$, respectively.

3.6 Exceptions: Schemes needing Direct Proofs

The only IBI scheme we found in the literature that is *not* based on a convertible SI scheme is the *OKDL-IBI* scheme shown in Figure 3.17 [Oka93]. It is based on discrete logarithms and is similar to the *Beth¹-IBI* scheme in that a user's secret key is a signature on his identity and the identification protocol is a proof of knowledge of such signature. We prove its security under the discrete logarithm assumption, a result missing from the original work [Oka93]. We also present a more natural and slightly more efficient variant of *OKDL-IBI* that we call the *XDL-IBI* scheme, and that just like the *OKDL-IBI* scheme is not based on a cSI scheme but can be proven secure directly as an IBI scheme.

3.6.1 The *OkDL-IBI* Scheme

Theorem 3.22 *The OkDL-IBI scheme is secure against impersonation under passive, active and concurrent attacks (*imp-pa/aa/ca *secure) in the random oracle model if the discrete logarithm problem associated to* $\mathsf{K}_{\mathrm{dlog}}$ *is hard.* ∎

Proof: We prove the above theorem by showing that if there exists a polynomial-time impersonator $\overline{\mathsf{A}}$ breaking *OkDL-IBI* in a concurrent attack using $\mathbf{Q}_{\overline{\mathsf{A}}}^{\mathrm{Init}}$ initialization queries and $\mathbf{Q}_{\overline{\mathsf{A}}}^{\mathrm{H}}$ queries to the random oracle, then there exists an algorithm B solving the discrete logarithm problem associated to $\mathsf{K}_{\mathrm{dlog}}$ such that

$$\mathbf{Adv}^{\text{imp-ca}}_{\textit{OkDL-IBI},\overline{\mathsf{A}}}(k) \leq c_1 \cdot \sqrt{\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},B}(k)} + c_2 \tag{3.8}$$

$$\text{where} \quad c_1 = 1 + \sqrt{(1 + \mathbf{Q}_{\overline{\mathsf{A}}}^{\mathrm{H}}) \cdot \tfrac{2^{k-1}}{2^{k-1}-1}}$$
$$c_2 = 2^{-l(k)} + \sqrt{2^{1-k} \cdot (1 + \mathbf{Q}_{\overline{\mathsf{A}}}^{\mathrm{Init}} + \mathbf{Q}_{\overline{\mathsf{A}}}^{\mathrm{H}}) \cdot \mathbf{Q}_{\overline{\mathsf{A}}}^{\mathrm{Init}}}\ .$$

A user's secret in the *OkDL-IBI* scheme is essentially an optimized[2] signature of the user's identity under a signature scheme that is commonly known as the classical Okamoto signature scheme [Oka93], that we refer to as the *OkCL-SS* scheme here. This scheme is the fs-I-2-S transform of the *OkCL-SI* scheme depicted in Figure 3.18. (Note that the *OkCL-SI* and *OkCL-SS* schemes are *not* convertible, so corresponding IBI and IBS schemes are not defined.)

As usual, the proof works by contradiction: given an imp-ca adversary $\overline{\mathsf{A}}$ for the *OkDL-IBI* scheme, we construct an algorithm B that is able to compute discrete logarithms in $\mathbb{G}$. We distinguish between two types of impersonations: the first reusing previously seen values for h, S in the attack, and the second creating its own values for h, S. We first show how to transform the former type into a discrete logarithm algorithm B_1 directly. For the second type of impersonation, we take a modular approach by proving it equivalent to breaking the *weak non-malleability* (to be introduced shortly hereafter) of the *OkCL-SS* scheme, which in turn is shown to be implied by the security of *OkCL-SI* under passive attack, which finally is known to hold under the discrete logarithm assumption. The cascade of algorithms in the reduction is illustrated in Figure 3.19.

We start off with the description of the discrete logarithm algorithm B_1. Given an imp-ca adversary $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ and input $(\mathbb{G}, q, g_1, g_2)$, it computes $\log_{g_1} g_2$ as follows. It chooses x_1, x_2 at random from $\mathbb{Z}_q$, computes $X \leftarrow g_1^{-x_1} g_2^{-x_2}$ and runs $\overline{\mathsf{CV}}$ on input $mpk = (\mathbb{G}, q, g_1, g_2, X)$. It answers all $\overline{\mathsf{CV}}$'s oracle queries by running the real *OkDL-IBI* protocol algorithms, which it can since it knows

[2] Instead of $Y \| z_1 \| z_2$, often the equivalent but more compact representation $h \| z_1 \| z_2$ with $h = H(Y \| M)$ is used as the signature, since Y can be recomputed as $g_1^{z_1} g_2^{z_2} X^h$. It is this representation that is used as the user secret key in *OkDL-IBI*.

Algorithm $\mathsf{Kg}(1^k)$
$(\mathbb{G}, q, g_1) \xleftarrow{R} \mathsf{K}_{\mathrm{dlog}}(1^k)$
$\alpha \xleftarrow{R} \mathbb{Z}_q$; $g_2 \leftarrow g_1^{\alpha}$; $x_1, x_2 \xleftarrow{R} \mathbb{Z}_q$; $X \leftarrow g_1^{-x_1} g_2^{-x_2}$
$pk \leftarrow ((\mathbb{G}, q, g_1, g_2), X)$; $sk \leftarrow ((\mathbb{G}, q, g_1, g_2), (x_1, x_2))$
Return (pk, sk)

Prover P		Verifier V
$y_1, y_2 \xleftarrow{R} \mathbb{Z}_q$; $Y \leftarrow g_1^{y_1} g_2^{y_2}$	$\xrightarrow{Y}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \{0,1\}^{l(k)}$
$z_1 \leftarrow y_1 + cx_1 \bmod q$		
$z_2 \leftarrow y_2 + cx_2 \bmod q$	$\xrightarrow{z_1, z_2}$	
		If $Y \in \mathbb{G}$ and $z_1, z_2 \in \mathbb{Z}_q$ and $Y \equiv g_1^{z_1} g_2^{z_2} X^c$ then `acc` else `rej`

Figure 3.18: **The *OkCL-SI* scheme.** The scheme is parameterized by discrete-log group generator $\mathsf{K}_{\mathrm{dlog}}$ and super-logarithmic challenge length $l : \mathbb{N} \to \mathbb{N}$ such that $2^{l(k)} < q$ for all q output by $\mathsf{K}_{\mathrm{dlog}}(1^k)$. The prover P and verifier V are run on initial states $sk = ((\mathbb{G}, q, g_1, g_2), (x_1, x_2))$ and $pk = ((\mathbb{G}, q, g_1, g_2), X)$, respectively. The scheme is canonical but not convertible, so it can be transformed into the *OkCL-SS* $=$ fs-I-2-S(*OkCL-SI*) scheme, but no corresponding IBI or IBS schemes are defined.

the master secrets x_1, x_2, storing the user secret key it generates for each identity I as $(h_I, s_{1,I}, s_{2,I})$. At the end of its execution, $\overline{\mathsf{CV}}$ outputs the identity J that will be attacked and state information $St_{\overline{\mathsf{CP}}}$ for the cheating prover. For the remainder of this paragraph, we use $\tilde{h}$ as shorthand notation for h_J, $\tilde{s}_1$ for $s_{1,J}$, $\tilde{s}_2$ for $s_{2,J}$ and $\tilde{S}$ for $g_1^{-\tilde{s}_1} g_2^{-\tilde{s}_2}$, respectively. Consider a modified verifier algorithm $\overline{\mathsf{V}}'$ that only accepts conversation (h, S, Y, c, z_1, z_2) if $\overline{\mathsf{V}}$ would accept it and $(h, S) = (\tilde{h}, \tilde{S})$. Algorithm B_1 runs $\overline{\mathsf{CP}}$ in a reset experiment against this modified verifier $\overline{\mathsf{V}}'$, again simulating oracles by running the real *OkDL-IBI* algorithms, to generate two accepting conversations $(h, S, Y, c_1, z_{11}, z_{12}), (h, S, Y, c_2, z_{21}, z_{22})$ for randomly chosen challenges c_1, c_2. From these conversations, it is possible to extract s_1, s_2 such that $S \equiv g_1^{-s_1} g_2^{-s_2} \equiv \tilde{S}$ as $s_i \leftarrow (c_1 - c_2)^{-1}(z_{1i} - z_{2i}) \bmod q$. The combined views of $\overline{\mathsf{CV}}$ and $\overline{\mathsf{CP}}$ are independent of B_1's choice for $(\tilde{s}_1, \tilde{s}_2)$, so with probability $(q-1)/q$ we have $(s_1, s_2) \neq (\tilde{s}_1, \tilde{s}_2)$ and the discrete logarithm of g_2 relative to g_1 can be computed as $-(s_1 - \tilde{s}_1)(s_2 - \tilde{s}_2)^{-1} \bmod q$.

The simulation of $\overline{\mathsf{CV}}$'s and $\overline{\mathsf{CP}}$'s environment is obviously perfect, since the same algorithms were used as in the real game. Let $\mathbf{E}$ be the event that

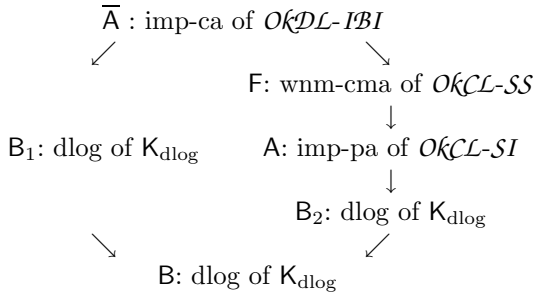


Figure 3.19: **Cascade of reductions in the proof of Theorem 3.22.** An arrow going from label "X_1: Y_1 of Z_1" to label "X_2: Y_2 of Z_2" means that the existence of a polynomial-time algorithm X_1 attacking scheme Z_1 under security notion Y_1 implies the existence of an algorithm X_2 breaking scheme Z_2 under notion Y_2.

$(\tilde{h}, \tilde{S}) = (h, S)$, and let $\overline{\mathbf{E}}$ be the complementary event. Let $acc'(St_{\overline{\mathsf{CP}}}, (mpk, J))$ be the probability that $\overline{\mathsf{V}}'$ accepts on input (mpk, J) after interacting with $\overline{\mathsf{CP}}$ initiated on $St_{\overline{\mathsf{CP}}}$, and let $res'(St_{\overline{\mathsf{CP}}}, (mpk, J)$ be the probability that the reset experiment confronting $\overline{\mathsf{CP}}$ with $\overline{\mathsf{V}}'$ as in Lemma 3.15 returns 1. We have the following expression for the advantage of B_1 in computing discrete logarithms:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}_1}(k) &\geq res'(St_{\overline{\mathsf{CP}}}, (mpk, J)) \cdot \Pr[\mathbf{E}] \\ &\geq \left(acc'(St_{\overline{\mathsf{CP}}}, (mpk, J)) - 2^{-l(k)}\right)^2 \cdot \Pr[\mathbf{E}]\end{aligned}$$

This concludes the description of algorithm B_1.

We now focus on how the second type of attack that creates new values (h, S) for its impersonation can be transformed into a discrete logarithm algorithm. The $\mathcal{OKCL}$-$\mathcal{SI}$ scheme is known to be secure against impersonation under concurrent attack under the discrete logarithm assumption, and hence by Theorem 3.4 the $\mathcal{OKCL}$-$\mathcal{SS}$ scheme is unforgeable under chosen-message attack under the same assumption in the random oracle model. This, however, is not sufficient for our purposes, since a forger F will only be able to extract a second signature (h, s_1, s_2) on the previously signed message J. Inspired by the notion of *non-malleability* [SPMLS02], we say that a signature scheme $\mathcal{SS} = \mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{SI}) = (\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$ associated to a canonical SI scheme $\mathcal{SI}$ is *weakly non-malleable under chosen-message attack* if no polynomial-time algorithm F has non-negligible advantage in winning the following game:

Experiment $\mathbf{Exp}^{\text{wnm-cma}}_{\mathcal{SS},\mathsf{F}}(k)$
$(pk, sk) \xleftarrow{R} \mathsf{Kg}(1^k)$

$(M, Cmt\|Rsp) \leftarrow \mathsf{F}(pk : \textsc{Sign})$,
answering queries $\textsc{Sign}(M_i)$ as $Cmt_i\|Rsp_i \stackrel{R}{\leftarrow} \mathsf{Sign}(sk, M_i)$
If $\mathsf{Vf}(pk, M, Cmt\|Rsp) = 1$ and $\nexists\, i$ such that $M = M_i$ and $Cmt = Cmt_i$ then return 1 else return 0.

Before showing that the $\mathcal{OKCL}$-$\mathcal{SS}$ scheme is weakly non-malleable, we first explain how the second type of impersonation is transformed into an algorithm F breaking the weak non-malleability of $\mathcal{OKCL}$-$\mathcal{SS}$.

Given an imp-ca adversary $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$, input $pk = (\mathbb{G}, q, g_1, g_2, X)$ and access to a signing oracle $\textsc{Sign}(\cdot)$ and random oracle $\mathrm{H}(\cdot)$, F proceeds as follows. It runs $\overline{\mathsf{CV}}$ on input $mpk = (\mathbb{G}, q, g_1, g_2, X)$, answering its oracle queries as:

- $\textsc{Init}(I)$: by calling and storing $(h_I, s_{1,I}, s_{2,I}) \leftarrow \textsc{Sign}(I)$ and returning 1
- $\textsc{Prov}(I, s, M)$: by running the real prover algorithm (which it can because it knows the user secret keys of all identities)
- $\textsc{Corr}(I)$: by returning $(h_I, s_{1,I}, s_{2,I})$

until $\overline{\mathsf{CV}}$ outputs $(St_{\overline{\mathsf{CP}}}, J)$. We use the same shorthand notations $\tilde{h}, \tilde{s}_1, \tilde{s}_2, S$ here as in the description of the discrete logarithm algorithm B_1 above. Define a modified verifier algorithm $\overline{\mathsf{V}}''$ that accepts only if $\overline{\mathsf{V}}$ accepts and moreover $(\tilde{h}, \tilde{S}) \neq (h, S)$. Algorithm F runs $\overline{\mathsf{CP}}$ in a reset experiment against $\overline{\mathsf{V}}''$, and extracts s_1, s_2 such that $S \equiv g_1^{-s_1} g_2^{-s_2}$ exactly as done in the B_1 algorithm. Since $h = \mathrm{H}(X^h S\|J)$ and $(\tilde{h}, \tilde{S}) \neq (h, S)$, the tuple $(h, -s_1 \bmod q, -s_2 \bmod q)$ is a valid signature on message J different from the one output by the signing oracle, thereby breaking the weak non-malleability.

It is easy to see that the simulation of the environment for $\overline{\mathsf{CV}}, \overline{\mathsf{CP}}$ is perfect. Let $acc''(St_{\overline{\mathsf{CP}}}, (mpk, J))$ and $res''(St_{\overline{\mathsf{CP}}}, (mpk, J))$ be as defined in the Reset Lemma when $\overline{\mathsf{CP}}$ is confronted with verifier $\overline{\mathsf{V}}''$. The advantage of F in breaking the weak non-malleability of the $\mathcal{OKCL}$-$\mathcal{SS}$ scheme is bounded by

$$\begin{aligned} \mathbf{Adv}^{\text{wnm-cma}}_{\mathcal{OKCL}\text{-}\mathcal{SS},\mathsf{F}}(k) &\geq res''(St_{\overline{\mathsf{CP}}}), (mpk, J)) \cdot \Pr[\overline{\mathbf{E}}] \\ &\geq \left(acc''(St_{\overline{\mathsf{CP}}}), (mpk, J)) - 2^{-l(k)}\right)^2 \cdot \Pr[\overline{\mathbf{E}}], \end{aligned}$$

and the running time and number of oracle queries of F are given by

$$\mathbf{T}_{\mathsf{F}} = O(\mathbf{T}_{\overline{\mathsf{A}}}), \quad \mathbf{Q}^{\textsc{Sign}}_{\mathsf{F}} = \mathbf{Q}^{\textsc{Init}}_{\overline{\mathsf{A}}}, \quad \mathbf{Q}^{\mathrm{H}}_{\mathsf{F}} = \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{A}}}\ . \tag{3.9}$$

Let $acc(St_{\overline{\mathsf{CP}}}, (mpk, J))$ be the probability that verifier $\overline{\mathsf{V}}$ accepts after interacting with $\overline{\mathsf{CP}}$. Observe that in event E, verifier $\overline{\mathsf{V}}'$ is equivalent to $\overline{\mathsf{V}}$, and likewise in event $\overline{\mathbf{E}}$, verifier $\overline{\mathsf{V}}''$ is equivalent to $\overline{\mathsf{V}}$, so by combining the equations

for the advantages of B and F, we can upper bound the advantage of $\overline{\mathsf{A}}$ as

$$\begin{aligned}
\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{OKDL}\text{-}\mathcal{IBI},\overline{\mathsf{A}}}(k) &= acc(St_{\overline{\mathsf{CP}}}, (mpk, J)) \\
&= acc'(St_{\overline{\mathsf{CP}}}, (mpk, J)) \cdot \Pr[\mathbf{E}] \\
&\quad + acc''(St_{\overline{\mathsf{CP}}}, (mpk, J)) \cdot \Pr[\overline{\mathbf{E}}] \\
&\leq \sqrt{\mathbf{Adv}^{\text{dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{B}_1}(k) \cdot \Pr[\mathbf{E}]} + \sqrt{\mathbf{Adv}^{\text{wnm-cma}}_{\mathcal{OKCL}\text{-}\mathcal{SS},\mathsf{F}}(k) \cdot \Pr[\overline{\mathbf{E}}]} \\
&\quad + 2^{-l(k)} \cdot (\Pr[\mathbf{E}] + \Pr[\overline{\mathbf{E}}]) \\
&\leq \sqrt{\mathbf{Adv}^{\text{dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{B}_1}(k)} + \sqrt{\mathbf{Adv}^{\text{wnm-cma}}_{\mathcal{OKCL}\text{-}\mathcal{SS},\mathsf{F}}(k)} + 2^{-l(k)}. \quad (3.10)
\end{aligned}$$

At this point, we have shown that in order to break the *OKDL-IBI* scheme, one has to be able to either compute discrete logarithms, or break the weak non-malleability of *OKCL-SS*. We now proceed to prove that the latter is equivalent to computing discrete logarithms as well.

Lemma 3.23 Let $\mathcal{SI}$ be a non-trivial canonical SI scheme with commitments drawn from CmtSet, and let $\mathcal{SS} = \mathsf{fs\text{-}I\text{-}2\text{-}S}(\mathcal{SI})$ as per Construction 3.3. If $\mathcal{SI}$ is secure against impersonation under passive attack, then $\mathcal{SS}$ is weakly non-malleable under chosen-message attack in the random oracle model. Moreover, if F is an algorithm breaking the weak non-malleability of $\mathcal{SS}$ using $\mathbf{Q}^{\text{Sign}}_{\mathsf{F}}$ sign-oracle queries and $\mathbf{Q}^{\text{H}}_{\mathsf{F}}$ queries to the random oracle, then there exists a passive impersonator A attacking $\mathcal{SI}$ such that

$$\mathbf{Adv}^{\text{wnm-cma}}_{\mathcal{SS},\mathsf{F}}(k) \leq (1+\mathbf{Q}^{\text{H}}_{\mathsf{F}}) \cdot \mathbf{Adv}^{\text{imp-pa}}_{\mathcal{SI},\mathsf{A}}(k) + \frac{(1 + \mathbf{Q}^{\text{H}}_{\mathsf{F}} + \mathbf{Q}^{\text{Sign}}_{\mathsf{F}}) \cdot \mathbf{Q}^{\text{Sign}}_{\mathsf{F}}}{|\text{CmtSet}|} \quad (3.11)$$

with $\mathbf{T}_{\mathsf{A}} = O(\mathbf{T}_{\mathsf{F}})$ and $\mathbf{Q}^{\text{Conv}}_{\mathsf{A}} = \mathbf{Q}^{\text{Sign}}_{\mathsf{F}}$. ∎

Proof: The description of algorithm A is identical to the impersonator described in the proof of Lemma 3.5 of Abdalla et al. [AABN02] (of which our Theorem 3.4 is a special case). In a nutshell, A uses the forger F as a subroutine to impersonate itself as a prover to an honest verifier V as follows. Algorithm A uses its conversation oracle to reply to F's signing and hash queries, except for one hash query $\text{H}(Cmt \| M)$ that it guesses to be the "crucial" query that F will use later in its forgery. When this query occurs, A sends Cmt as the first move of its identification to V, and returns the challenge it received from V as the response to F's hash query. If at the end F indeed outputs a valid forgery $Cmt \| Rsp$ for message M, then A successfully completes the identification protocol by sending Rsp as the response to V.

It is important that when the crucial hash query occurs, A is still free to program the value that will be returned to F. We can assume without loss of

generality that F never queries the hash oracle on the same argument twice, but the hash value might also have been fixed by a previous signature query for message M. At this point in the proof, Abdalla et al. [AABN02] exploit the fact that F is not allowed to make such signature query if it later wants to forge a signature on M. Here, we observe here that even if F retrieved a signature $Cmt'\|Rsp'$ for message M from the signing oracle before, then the value of $\mathrm{H}(Cmt\|M)$ is still undecided as long as $Cmt \neq Cmt'$, and this is exactly what is enforced by our definition of weak non-malleability. The rest of the analysis is the same as that of Abdalla et al. [AABN02], resulting in an almost identical advantage equation. ∎

Lemma 3.24 The *OkCL-SI* standard identification scheme associated to $\mathsf{K}_{\mathrm{dlog}}$ as depicted in Figure 3.18 is secure against forgery under passive attack if the discrete logarithm problem associated to $\mathsf{K}_{\mathrm{dlog}}$ is hard. Moreover, if A is a polynomial-time impersonator for *OkCL-SI* under passive attack, then there exists a polynomial-time algorithm B_2 for computing discrete logarithms such that

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathit{OkCL\text{-}SI},\mathsf{A}}(k) \;\leq\; \frac{2^{k-1}}{2^{k-1}-1} \cdot \mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}_2}(k).$$

∎

Proof: The description of algorithm B_2 is well-known [Oka93], so we restrict ourselves to proving Equation (3.24) here. The only time that algorithm B_2 fails while A succeeded is when the representation that B_2 extracts from A is exactly the one it already knew. Since there are q possible choices for B_2's representation and A's view is independent of this choice, the probability for this to happen is only $1/q$. Because the output of $\mathsf{K}_{\mathrm{dlog}}(1^k)$ is such that $2^{k-1} \leq q < 2^k$, it follows that

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}_2}(k) &\geq \left(1-\frac{1}{q}\right) \cdot \mathbf{Adv}^{\text{imp-pa}}_{\mathit{OkCL\text{-}SI},\mathsf{A}}(k) \\
&\geq \left(1-\frac{1}{2^{k-1}}\right) \cdot \mathbf{Adv}^{\text{imp-pa}}_{\mathit{OkCL\text{-}SI},\mathsf{A}}(k)
\end{aligned}$$

from which Equation (3.24) follows. ∎

Combining Equations 3.9, 3.10, 3.11 and 3.24 gives

$$\begin{aligned}
\mathbf{Adv}^{\text{imp-ca}}_{\mathit{OkDL\text{-}IBI},\overline{\mathsf{A}}}(k) &\leq \sqrt{\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}_1}(k)} \;+ 2^{-l(k)} \\
&+ \sqrt{(1+\mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{A}}}) \cdot \frac{2^{k-1}}{2^{k-1}-1} \cdot \mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}_2}(k) + \frac{(1+\mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{A}}}+\mathbf{Q}^{\mathrm{INIT}}_{\overline{\mathsf{A}}}) \cdot \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{A}}}}{2^{k-1}}}. \quad (3.12)
\end{aligned}$$

Now consider discrete logarithm algorithm B that, on input $(\mathbb{G}, q, g_1, g_2)$, runs B_1 on the same input if $\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},B_1}(|q|) \geq \mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},B_2}(|q|)$, and runs B_2 otherwise. Then for all $k \in \mathbb{N}$ we have

$$\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}}(k) = \max\left(\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}_1}(k),\ \mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}_2}(k)\right) .$$

Substituting this in Equation (3.12) and using the fact that $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ for all positive reals x, y yields Equation (3.8), as required. This concludes the proof of Theorem 3.22. ∎

As already noted in Section 3.4.2, the security of the *OkDL-IBS* scheme is *not* implied by Corollary 3.11 since the corresponding SI scheme is not convertible. The extended fs-I-2-S transform however does convert *OkDL-IBI* into an IBS scheme that is uf-cma secure in the random oracle model under the discrete logarithm assumption. The proof of the following theorem is a simple combination of Theorems 3.22 and 3.14.

Theorem 3.25 *The OkDL-IBS* = efs-IBI-2-IBS(*OkDL-IBI*) *as per Figure 3.17 and Construction 3.13 is unforgeable under chosen-message attack in the random oracle model if the discrete logarithm problem associated to* $\mathsf{K}_{\mathrm{dlog}}$ *is hard.* ∎

3.6.2 The *XDL-IBI* Scheme

We introduce a new IBI scheme called *XDL-IBI* that can be viewed as the single-generator variant of the *OkDL-IBI* scheme. We present it in Figure 3.20. It is provably secure as an IBI scheme yet is not the transformation of a convertible SI scheme. Just like the *OkDL* scheme, the uf-cma security of fs-I-2-S(*XDL-IBI*) is not implied by Corollary 3.11, yet a secure IBS scheme can be constructed using the extended efs-IBI-2-IBS transformation.

Theorem 3.26 *The XDL-IBI scheme is secure against impersonation under passive attack (*imp-pa *secure) in the random oracle model if the discrete logarithm problem associated to* $\mathsf{K}_{\mathrm{dlog}}$ *is hard.* ∎

Proof: We prove the theorem by showing that if there exists a polynomial-time impersonator $\overline{\mathsf{A}}$ breaking *XDL-IBI* in a passive attack using $\mathbf{Q}^{\mathrm{INIT}}_{\overline{\mathsf{A}}}$ initialization queries and $\mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{A}}}$ queries to the random oracle, then there exists an algorithm B

Algorithm $\mathsf{MKg}(1^k)$	Algorithm $\mathsf{UKg}(msk, I : \mathrm{H})$
$(\mathbb{G}, q, g) \xleftarrow{R} \mathsf{K}_{\mathrm{dlog}}(1^k)$	$(\mathbb{G}, q, g, x) \leftarrow msk$
$x \xleftarrow{R} \mathbb{Z}_q \; ; \; X \leftarrow g^x$	$r \xleftarrow{R} \mathbb{Z}_q \; ; \; R \leftarrow g^r$
$mpk \leftarrow (\mathbb{G}, q, g, X)$	$h \leftarrow \mathrm{H}(R \| I) \; ; \; s \leftarrow r + hx \bmod q$
$msk \leftarrow (\mathbb{G}, q, g, x)$	$usk \leftarrow (\mathbb{G}, q, g, h, s)$
Return (mpk, msk)	Return usk

Prover $\overline{\mathsf{P}}$		Verifier $\overline{\mathsf{V}}$
$S \leftarrow g^s$		
$y \xleftarrow{R} \mathbb{Z}_q \; ; \; Y \leftarrow g^y$	$\xrightarrow{h, S, Y}$	
	$\xleftarrow{c}$	$c \xleftarrow{R} \{0,1\}^{l(k)}$
$z \leftarrow y + cs \bmod q$	$\xrightarrow{z}$	
		If $h \in \{0,1\}^{l(k)}$ and $S, Y \in \mathbb{G}$ and $z \in \mathbb{Z}_q$ and $g^z \equiv Y S^c$ and $h = \mathrm{H}(SX^{-h} \| I)$ then `acc` else `rej`

Figure 3.20: **The $\mathcal{XDL}$-$\mathcal{IBI}$ scheme.** The scheme is parameterized by discrete-log group generator $\mathsf{K}_{\mathrm{dlog}}$ and superlogarithmic challenge length $l : \mathbb{N} \to \mathbb{N}$ such that $2^{l(k)} < q$ for all q output by $\mathsf{K}_{\mathrm{dlog}}(1^k)$. The prover $\overline{\mathsf{P}}$ and verifier $\overline{\mathsf{V}}$ are run on initial states $usk = (\mathbb{G}, q, g, h, s)$ and (mpk, I) where $mpk = (\mathbb{G}, q, g, X)$, respectively.

solving the discrete logarithm problem associated to $\mathsf{K}_{\mathrm{dlog}}$ such that

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{XDL}\text{-}\mathcal{IBI},\overline{\mathsf{A}}}(k) \leq c_1 \cdot \sqrt[4]{\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}}(k)} + c_2 \tag{3.13}$$

$$\text{where} \quad c_1 = \sqrt{(1 + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{A}}} + \mathbf{Q}^{\mathrm{Init}}_{\overline{\mathsf{A}}})}$$

$$c_2 = 2^{-l(k)} + \sqrt{(1 + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{A}}}) \cdot 2^{-l(k)} + \frac{(1 + \mathbf{Q}^{\mathrm{H}}_{\overline{\mathsf{A}}} + \mathbf{Q}^{\mathrm{Init}}_{\overline{\mathsf{A}}}) \cdot \mathbf{Q}^{\mathrm{Init}}_{\overline{\mathsf{A}}}}{2^{k-1}}} \; .$$

Similarly to the proof of Theorem 3.22, we prove the theorem by transforming any imp-pa adversary $\overline{\mathsf{A}} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ into a discrete logarithm algorithm B_1 and an algorithm F breaking the weak non-malleability of the $\mathcal{Schnorr}$-$\mathcal{SS}$ = fs-I-2-S($\mathcal{Schnorr}$-$\mathcal{SI}$) signature scheme [Sch90] obtained by applying latter algorithm is a rather straightforward adaptation of algorithm F in the proof of Theorem 3.22, the former needs a little more explanation.

On input $(\mathbb{G}, q, g_1, g_2)$, algorithm B_1 chooses $x \xleftarrow{R} \mathbb{Z}_q$, computes $X \leftarrow g_1^x$ and runs $\overline{\mathsf{CV}}$ on input $(\mathbb{G}, q, g_1, X)$. B_1 also chooses $q_{\mathrm{guess}} \xleftarrow{R} \{1, \ldots, \mathbf{Q}^{\mathrm{Init}}_{\overline{\mathsf{CV}}}\}$ and hopes that the identity I_{guess} initialized in the q_{guess}-th Init query will be the

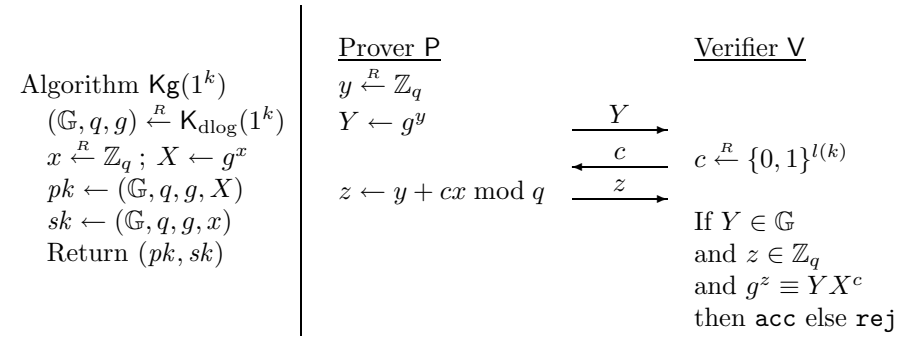


Figure 3.21: **The *Schnorr-SI* scheme.** The scheme is parameterized by discrete-log group generator $\mathsf{K}_{\mathrm{dlog}}$ and super-logarithmic challenge length $l : \mathbb{N} \to \mathbb{N}$ such that $2^{l(k)} < q$ for all q output by $\mathsf{K}_{\mathrm{dlog}}(1^k)$. The prover P and verifier V are run on initial states $sk = (\mathbb{G}, q, g, x)$ and $pk = (\mathbb{G}, q, g, X)$, respectively.

one under attack in the second phase of the game. All $\textsc{Init}(\cdot)$, $\textsc{Conv}(\cdot)$ and $\textsc{Corr}(\cdot)$ oracle queries are simulated using the real protocol algorithms, except for queries involving identity I_{guess}. When I_{guess} is initialized, B_1 chooses $\tilde{h} \xleftarrow{R} \{0,1\}^{l(k)}$, lets $\tilde{S} \leftarrow g_2$ and computes $\tilde{R} \leftarrow \tilde{S}X^{-\tilde{h}}$. Because until now $\overline{\mathsf{CV}}$'s view is independent of $\tilde{R}$, with overwhelming probability the random oracle $\mathrm{H}(\cdot)$ will not have been queried on value $\tilde{R}\|I_{\mathrm{guess}}$ before, so that B_1 can set $\mathrm{H}(\tilde{R}\|I_{\mathrm{guess}}) = \tilde{h}$. Conversations for I_{guess} are generated by choosing $c \xleftarrow{R} \{0,1\}^{l(k)}$, $z \xleftarrow{R} \mathbb{Z}_q$, computing $Y \leftarrow g_1^z \tilde{S}^{-c}$ and returning conversation $(\tilde{h}, \tilde{S}, Y, c, z)$. In the second stage of the game, $\overline{\mathsf{CP}}$ is run in a reset experiment to generate two accepting conversations (h, S, Y, c_1, z_1), (h, S, Y, c_2, z_2) against a modified verifier $\overline{\mathsf{V}}'$ that only accepts if $\overline{\mathsf{V}}$ does and $(h, S) = (\tilde{h}, \tilde{S})$. The discrete logarithm of g_2 with respect to g_1 is then computed as $(c_1 - c_2)^{-1}(z_1 - z_2) \bmod q$.

By an analysis similar to that in the proof of Theorem 3.22, the imp-pa advantage of $\overline{\mathsf{A}}$ can be bounded by

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{XDL}\text{-}\mathcal{IBI},\overline{\mathsf{A}}}(k) \leq \sqrt{\mathbf{Q}^{\textsc{Init}}_{\overline{\mathsf{CV}}} \cdot \mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},\mathsf{B}_1}(k)} + \sqrt{\mathbf{Adv}^{\text{wnm-cma}}_{\mathcal{Schnorr}\text{-}\mathcal{SS},\mathsf{F}}(k)} + 2^{-l(k)},$$

where the second term can be reduced further through Lemma 3.23 and

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{Schnorr}\text{-}\mathcal{SI},(}(k) \leq \sqrt{\mathbf{Adv}^{\mathrm{dlog}}_{\mathsf{K}_{\mathrm{dlog}},B_2}(k)} + 2^{-l(k)}$$

to yield Equation (3.13) as required. Note that the fourth root in Equation (3.13) is induced by the reset experiment that, unlike the $\mathcal{OKCL}\text{-}\mathcal{SI}$, is needed in the reduction of computing discrete logarithms to breaking the imp-pa security of the *Schnorr-SI* scheme. ∎

The following theorem then follows easily from Theorem 3.14.

Theorem 3.27 *The $\mathcal{XDL}$-$\mathcal{IBS}$ = $\mathsf{efs\text{-}IBI\text{-}2\text{-}IBS}(\mathcal{XDL}\text{-}\mathcal{IBS})$ scheme as defined by Figure 3.20 and Construction 3.13 is unforgeable under chosen-message attack in the random oracle model if the discrete logarithm problem associated to K_{dlog} is hard.* ∎

It is unknown if the $\mathcal{XDL}$-$\mathcal{IBI}$ is also secure against impersonation under active and concurrent attacks under the plain discrete logarithm assumption. A proof does exist however under the stronger one-more discrete logarithm assumption.

Theorem 3.28 *The $\mathcal{XDL}$-$\mathcal{IBI}$ scheme is secure against impersonation under concurrent attacks (*imp-ca*) in the random oracle model if the one-more discrete logarithm problem associated to K_{dlog} is hard.* ∎

Proof: We will show how to, given a polynomial-time impersonator $\overline{\mathsf{A}}$ breaking $\mathcal{XDL}$-$\mathcal{IBI}$ under concurrent attack, build an algorithm B solving the one-more discrete logarithm problem associated to K_{dlog} such that

$$\mathbf{Adv}^{\text{imp-pa}}_{\mathcal{XDL}\text{-}\mathcal{IBI},\overline{\mathsf{A}}}(k) \leq c_1 \cdot \sqrt[4]{\mathbf{Adv}^{\text{1m-dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{B}}(k)} + c_2 \tag{3.14}$$

$$\text{where} \quad c_1 = 1 + \sqrt{(1 + \mathbf{Q}^{\text{H}}_{\overline{\mathsf{A}}})}$$

$$c_2 = 2^{-l(k)} + \sqrt{(1 + \mathbf{Q}^{\text{H}}_{\overline{\mathsf{A}}}) \cdot 2^{-l(k)} + \frac{(1 + \mathbf{Q}^{\text{H}}_{\overline{\mathsf{A}}} + \mathbf{Q}^{\text{Init}}_{\overline{\mathsf{A}}}) \cdot \mathbf{Q}^{\text{Init}}_{\overline{\mathsf{A}}}}{2^{k-1}}}$$

The one-more discrete logarithm problem associated to K_{dlog} is to, given $(\mathbb{G}, g, q)$ and access to challenge oracle $\textsc{Chall}(\cdot)$ and discrete logarithm oracle $\textsc{DLog}(\cdot)$, output the discrete logarithm of all target points received from the $\textsc{Chall}$ oracle using strictly less $\textsc{DLog}$ queries.

We use the same approach once again by reducing an imp-ca adversary $\mathsf{A} = (\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ into a one-more discrete logarithm algorithm B_1 and an algorithm F breaking the weak non-malleability of $\mathcal{Schnorr}$-$\mathcal{SS}$ signatures. The latter algorithm is identical to that in the proof of Theorem 3.26, but using the user secret keys to simulate interactive prover sessions instead of conversations.

Algorithm B_1 uses the challenge oracle to produce values S_I for all identities I initialized by $\overline{\mathsf{A}}$ and simulates interactive prover sessions by retrieving $Y_{I,i} \leftarrow \textsc{Chall}(\varepsilon)$ and computing the response for challenge $c_{I,i}$ as $z_{I,i} \leftarrow \textsc{DLog}(Y_{I,i} S_I^{c_{I,i}})$. When $\overline{\mathsf{A}}$ announces to break identity J and proceeds to the second phase of the game, B_1 runs a reset experiment to extract the discrete logarithm $\tilde{s}$ of $\tilde{S} = S_J$ and uses it to compute discrete logarithms of all values $Y_{J,i}$ as $y_{J,i} \leftarrow z_{J,i} - \tilde{s} c_{J,i} \bmod q$. For all other initialized identities $I \neq J$, B_1 asks

for the discrete logarithm $s_I \leftarrow \text{DLOG}(S_I)$ itself and computes the discrete logarithms $y_{I,i} \leftarrow z_{I,i} - s_I c_{I,i} \bmod q$. Let n be the number of identities initialized by $\overline{\mathsf{A}}$, and let n_I be the number of prover sessions initiated for identity I. Then for each identity I, B_1 calculated the discrete logarithm of $n_I + 1$ target points (all $Y_{I,i}$ and S_I) using $n_I + 1$ queries to the DLOG oracle (one for each prover session, and an additional one at the end of the game), *except* for J where the discrete logarithms of $n_J + 1$ target points were computed using only n_J queries to the DLOG oracle. So in total, B_1 saved one DLOG query and wins the game.

The advantage of an imp-ca adversary $\overline{\mathsf{A}}$ is bounded by

$$\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{XDL}\text{-}\mathcal{IBI},\overline{\mathsf{A}}}(k) \leq \sqrt{\mathbf{Adv}^{\text{1m-dlog}}_{\mathsf{K}_{\text{dlog}},\mathsf{B}_1}(k)} + \sqrt{\mathbf{Adv}^{\text{wnm-cma}}_{\mathcal{Schnorr}\text{-}\mathcal{SS},\mathsf{F}}(k)} + 2^{-l(k)}$$

which by similar techniques as those used in the proof of Theorem 3.26 yields Equation (3.14) as desired. ∎

3.7 Conclusion

In this chapter, we provided security proofs for existing and new identity-based identification and signatures schemes. We first extended the notions of security against impersonation under passive, active and concurrent attack for SI schemes to the identity-based setting, thereby filling a somewhat surprising gap viewing the large number of identity-based identification schemes proposed in the literature [FS86, GQ89, Oka93, Gir90, Bet88].

We then presented a framework of existing and new security-preserving transformations between (certain classes of) SI, SS, IBI and IBS schemes. The framework reduces proving security of IBI and IBS schemes to proving an underlying SI scheme, which is a considerably easier task. We applied the framework to 13 schemes proposed in the literature, thereby surfacing new related schemes and providing security proofs (or in one instance attacks) for SI schemes that were not analyzed before. Not only did our framework prove to be a valuable tool in proving the security of IBI and IBS schemes, but we also believe that it provides insight in now such schemes are constructed.

Finally, we discussed two exceptional IBI schemes that do not fall under our framework, but that we were able to prove secure as IBI schemes directly. The results of this chapter were published at EUROCRYPT 2004 [1].

Chapter 4

Transitive Signatures

4.1 Introduction and Main Contributions

We present novel realizations of the transitive signature (TS) primitive introduced by Micali and Rivest [MR02b], and also provide an answer to an open question they raise regarding the security of an RSA based scheme.

4.1.1 Background

THE CONCEPT. The context envisioned by Micali and Rivest [MR02b] is that of dynamically building an authenticated graph, edge by edge. The signer, having secret key *tsk* and public key *tpk*, can at any time pick a pair i, j of nodes and create a signature of $\{i, j\}$, thereby adding edge $\{i, j\}$ to the graph. A composability property is required: given a signature of an edge $\{i, j\}$ and a signature of an edge $\{j, k\}$, anyone in possession of the public key can create a signature of the edge $\{i, k\}$. Security asks that this limited class of forgeries be the only possible ones. (I.e., without *tsk*, it should be hard to create a valid signature of edge $\{i, j\}$ unless i, j are connected by a path whose edges have been explicitly authenticated by the signer.) Thus the authenticated graph at any point is the transitive closure of the graph formed by the edges explicitly authenticated by the signer, whence the name of the concept.

Applications suggested by Micali and Rivest [MR02b] include military chains-of-command, where nodes represent military personnel and a directed edge from i to j represents that i controls j, and administrative domains, where nodes represent machines and an undirected edge between i and j means that i and j are in the same domain. It seems that a truly compelling application, however, remains to be found. While such applications are more likely to be found for directed than for undirected graphs, (non-trivial) transitive signature schemes

for directed graphs appear to be much harder to construct: no schemes have been proposed so far, and following our work Hohenberger [Hoh03] provided evidence that either a new algebraic structure or a completely different approach is needed to do so. Our work focuses on undirected transitive signatures, and so from here on all graphs are assumed to be undirected.

Realizing the concept. A transitive signature scheme can be trivially realized by accepting, as a valid signature of $\{i, j\}$, any chain of signatures that authenticates a sequence of edges forming a path from i to j. Two issues lead [MR02b] to exclude this: the growth in signature size, and the loss of privacy incurred by having signatures carry information about their history. The main result of Micali and Rivest [MR02b] is a (non-trivial) transitive signature scheme, here denoted $\mathcal{DL}$-$\mathcal{TS}$, that is proven to be (transitively) unforgeable under adaptive chosen-message attack (see Section 4.2 for formal definitions) assuming that the discrete logarithm problem is hard in an underlying prime-order group and assuming security of an underlying standard signature scheme. They also present a natural RSA-based transitive signature scheme, here denoted $\mathcal{RSA}$-$\mathcal{TS}$, but point out that even though it seems secure, and a proof of transitive unforgeability under *non-adaptive* chosen-message attacks exists, there is no known proof of transitive unforgeability under *adaptive* chosen-message attacks. They thereby highlight the fact that in this domain, adaptive attacks might be harder to provably protect against than non-adaptive ones.

This work. In summary, prior to our work transitive signatures (transitively unforgeable under adaptive chosen-message attacks) had just a single realization, namely the $\mathcal{DL}$-$\mathcal{TS}$ scheme. It is standard practice in cryptography to seek new and alternative realizations of primitives of potential interest, both to provide firmer theoretical foundations for the existence of the primitive by basing it on alternative conjectured hard problems and to obtain performance improvements. In this chapter, we present new schemes that accomplish both of these objectives, and also provides an answer to the question about the $\mathcal{RSA}$-$\mathcal{TS}$ scheme.

The node certification technique. It is worth outlining the node certification based technique introduced by the $\mathcal{DL}$-$\mathcal{TS}$ scheme. The signer's keys include those of a standard digital signature scheme, and the public key includes additional items. (In the $\mathcal{DL}$-$\mathcal{TS}$ scheme, this is a group $\mathbb{G}$ of prime order q and a pair of generators of $\mathbb{G}$.) The signer associates to each node i in the current graph a *node certificate* consisting of a *public label* $L(i)$ and a signature on the concatenation of i and $L(i)$ under the standard scheme. The signature of an edge contains the certificates of its endpoints plus an *edge label* δ. Verification of an edge signature involves relating the edge label to the public labels of its endpoints as provided in the node certificates and verifying the standard signatures in the node certificates. Composition involves algebraic manipulation of

Scheme	Signing cost	Verification cost	Composition cost	Signature size	
$\mathcal{DL}$-$\mathcal{TS}$	2 stand. sigs 2 exp. in $\mathbb{G}$	2 stand. verifs 1 exp. in $\mathbb{G}$	2 adds in $\mathbb{Z}_q$	2 stand. sigs 2 points in $\mathbb{G}$ 2 points in $\mathbb{Z}_q$	4416 bits (SDL) 2708 bits (EC)
$\mathcal{DL1m}$-$\mathcal{TS}$	2 stand. sigs 1 exp. in $\mathbb{G}$	2 stand. verifs 1 exp. in $\mathbb{G}$	1 add in $\mathbb{Z}_q$	2 stand. sigs 2 points in $\mathbb{G}$ 1 point in $\mathbb{Z}_q$	4256 bits (SDL) 2548 bits (EC)
$\mathcal{RSA}$-$\mathcal{TS}$	2 stand. sigs 2 RSA encs	2 stand. verifs 1 RSA enc.	$O(\|N\|^2)$ ops	2 stand. sigs 3 points in $\mathbb{Z}_N^*$	5120 bits
$\mathcal{Fact}$-$\mathcal{TS}$	2 stand. sigs $O(\|N\|^2)$ ops	2 stand. verifs $O(\|N\|^2)$ ops	$O(\|N\|^2)$ ops	2 stand. sigs 3 points in $\mathbb{Z}_N^*$	5120 bits
$\mathcal{Gap}$-$\mathcal{TS}$	2 stand. sigs 2 exp. in $\hat{\mathbb{G}}$	2 stand. verifs 1 $\mathsf{S}_{\mathrm{ddh}}$	$O(\|N\|^2)$ ops	2 stand. sigs 3 points in $\hat{\mathbb{G}}$	2558 bits
$\mathcal{RSAH}$-$\mathcal{TS}$	1 RSA dec.	1 RSA enc.	$O(\|N\|^2)$ ops	1 point in $\mathbb{Z}_N^*$	1024 bits
$\mathcal{FactH}$-$\mathcal{TS}$	2 sq. roots in $\mathbb{Z}_N^*$	$O(\|N\|^2)$ ops	$O(\|N\|^2)$ ops	1 point in $\mathbb{Z}_N^*$	1024 bits
$\mathcal{GapH}$-$\mathcal{TS}$	1 exp. in $\hat{\mathbb{G}}$	1 $\mathsf{S}_{\mathrm{ddh}}$	$O(\|N\|^2)$ ops	1 point in $\hat{\mathbb{G}}$	170 bits

Figure 4.1: Cost comparisons amongst transitive signature schemes. The word "stand." refers to operations of the underlying standard signature scheme, which are eliminated for $\mathcal{RSAH}$-$\mathcal{TS}$, $\mathcal{FactH}$-$\mathcal{TS}$ and $\mathcal{GapH}$-$\mathcal{TS}$. $\mathbb{G}$ denotes the group of prime order q used in $\mathcal{DL}$-$\mathcal{TS}$ and $\mathcal{DL1m}$-$\mathcal{TS}$, and N denotes a modulus product of two primes as used in the RSA and factoring-based schemes. $\hat{\mathbb{G}}$ is a Gap Diffie-Hellman group and $\mathsf{S}_{\mathrm{ddh}}$ is an execution of the decision Diffie-Hellman algorithm in $\hat{\mathbb{G}}$. Abbreviations used are: "exp." for an exponentiation in the group; "RSA enc." for an RSA encryption; "RSA dec." for an RSA decryption performed given the decryption exponent; "sq. root" for a square root modulo N performed using the prime factors of N; and "ops" for the number of elementary bit operations in big-O notation. The final column gives approximate total signature lengths using 1024-bit RSA as standard signature scheme and with separate indications for subgroup discrete logarithm groups (SDL) and elliptic curve groups (EC).

edge labels.

The technique is useful, but brings an associated cost. Producing a signature for an edge can involve computing two normal signatures. The length of an edge signature, containing two node certificates each including a standard signature, can be large even if the edge labels are small.

4.1.2 Transitive Signatures based on RSA

This scheme, briefly mentioned by Micali and Rivest [MR02b], employs the node certification technique. The signer has keys for a standard signature scheme. Its public key additionally includes an RSA modulus N and encryption exponent e, while its secret key includes the corresponding decryption exponent d. The public label of a node i is a point $L(i) \in \mathbb{Z}_N^*$, and the edge label of edge $\{i, j\}$ is $L(i)^d L(j)^{-d} \bmod N$ assuming $i < j$. Composition involves multiplying edge labels modulo N. One can prove that $\mathcal{RSA}$-$\mathcal{TS}$ is transitively unforgeable under *non-adaptive* chosen-message attacks assuming the one-wayness of RSA and the security of the underlying standard signature scheme. No adaptive chosen-message attack that succeeds in forgery has been found, but neither has it been proven that $\mathcal{RSA}$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attack.

This situation (namely a scheme that appears to resist both attack and proof) is not uncommon in cryptography, and we suggest that it is a manifestation of the fact that the security of the scheme is relying on properties possessed by RSA but going beyond those captured by the assumption that RSA is one-way. Accordingly we seek an alternative, stronger assumption upon which a proof of security can be based.

We prove that $\mathcal{RSA}$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attacks under the assumption that the one-more RSA problem is hard (see Section 2.3.1) and that the standard signature scheme is secure.

4.1.3 New Transitive Signature Schemes

The $\mathcal{Fact}$-$\mathcal{TS}$ scheme. After seeing the $\mathcal{RSA}$-$\mathcal{TS}$ scheme, one might wonder whether there exists a transitive signature scheme that is provably secure (transitively unforgeable under adaptive chosen-message attack) under the standard one-wayness of RSA. We answer this question in a positive way by presenting the $\mathcal{Fact}$-$\mathcal{TS}$ scheme that is provably secure under the (even weaker) factoring assumption.

In our $\mathcal{Fact}$-$\mathcal{TS}$ scheme, the signer has keys for a standard signature scheme, and its public key additionally includes a modulus N that is the product of two large primes. The public label of a node i is a quadratic residue $L(i) \in \mathbb{Z}_N^*$, and an edge label of edge $\{i, j\}$ is a square root of $L(i)L(j)^{-1} \bmod N$ assum-

ing $i < j$. Composition involves multiplying edge labels modulo N. We prove that *Fact-TS* is transitively unforgeable under adaptive chosen-message attack, assuming the hardness of factoring the underlying modulus, and assuming security of the underlying standard signature scheme. The delicate part of this proof is an information-theoretic lemma showing that, even under an adaptive chosen-message attack, for any $\{i, j\}$ not in the transitive closure of the current graph, an adversary has zero advantage in determining which of the square roots of $L(i)L(j)^{-1}$ is held by the signer.

One might wonder why proofs under standard assumptions exist for *DL-TS* and *Fact-TS* but remain elusive for *RSA-TS* in spite of the obvious similarities between these schemes. The proofs for *DL-TS* and *Fact-TS* exploit the fact that there are multiple valid edge labels for any given edge in the graph, and that finding two different edge labels implies solving the underlying hard problem. With *RSA-TS*, the edge label is uniquely determined by the two node certificates, and this technique fails.

With regard to costs, we are interested in the computational cost of signing an edge (in the worst case that both endpoints of the edge are not in the current graph); the computational cost of verifying a candidate signature of an edge; the computational cost of composing two edge signatures to obtain another; and the size of a signature. Figures 4.1 and 4.2 summarize, respectively, the costs and provable-security attributes of the various schemes we have introduced, and compare them with the *DL-TS* scheme.

Since *Fact-TS* continues to employ the node certification technique, it incurs the same costs as *DL-TS* and *RSA-TS* from the use of the standard signature scheme. However, as Figure 4.1 indicates, it is otherwise computationally cheaper than *DL-TS* and *RSA-TS* for signing and verifying, reducing the extra cost from cubic (exponentiation) to quadratic (a couple of multiplications and an inverse).

THE *DL1m-TS* SCHEME. The *DL-TS* scheme [MR02b] uses two generators. We briefly note a simpler and perhaps more natural discrete-log based scheme called *DL1m-TS* that uses a single generator. This scheme is a discrete-log based analog of *RSA-TS*. As Figure 4.2 indicates, it offers some slight performance improvements over *DL-TS*. However, while the security of *DL-TS* is proven under the standard discrete-logarithm assumption [MR02b], our proof of security of *DL1m-TS* requires a stronger assumption, namely the hardness of the one-more discrete logarithm problem as defined in Section 2.3.2.

The tradeoff here is analogous to one arising for discrete-logarithm based SI schemes. *DL-TS* is similar to Okamoto's two-generator using SI scheme *OkCL-SI* [Oka93] presented in Figure 3.18, while *DL1m-TS* is similar to Schnorr's one-generator using *Schnorr-SI* scheme [Sch90] depicted in Figure 3.21. Schnorr's scheme is simpler, more natural and slightly more efficient. However, while Okamoto proved his scheme secure (against impersonation under active at-

Scheme	Proven to be transitively unforgeable under *adaptive* chosen-message attack assuming	RO?
$\mathcal{DL}$-$\mathcal{TS}$	Security of standard signature scheme Hardness of discrete logs in prime-order group	No
$\mathcal{DL1m}$-$\mathcal{TS}$	Security of standard signature scheme Hardness of one-more discrete logs in prime-order group	No
$\mathcal{RSA}$-$\mathcal{TS}$	Security of standard signature scheme RSA is secure against one-more-inversion attack	No
$\mathcal{F}act$-$\mathcal{TS}$	Security of standard signature scheme Hardness of factoring	No
$\mathcal{G}ap$-$\mathcal{TS}$	Security of standard signature scheme One-more gap Diffie-Hellman assumption	No
$\mathcal{RSAH}$-$\mathcal{TS}$	RSA is secure against one-more-inversion attack	Yes
$\mathcal{F}act\mathcal{H}$-$\mathcal{TS}$	Hardness of factoring	Yes
$\mathcal{G}ap\mathcal{H}$-$\mathcal{TS}$	One-more Gap Diffie-Hellman assumption	Yes

Figure 4.2: Provable security attributes of transitive signature schemes. We indicate the assumptions under which there is a proof of transitive unforgeability under *adaptive* chosen-message attack, and whether or not the random oracle model is used.

tack) under the standard discrete-logarithm assumption, the proof of security for Schnorr's scheme (which remained elusive for a while) is based on the hardness of the one-more discrete logarithm problem [BP02].

The $\mathcal{G}ap$-$\mathcal{TS}$ scheme. Gap Diffie-Hellman groups are groups where the CDH (Computational Diffie-Hellman) problem is hard but the DDH (Decision Diffie-Hellman) problem is easy (see Section 2.3.2 for more details). They have been used to yield short signatures [BLS01] and also simple, efficient schemes for threshold, blind and multi-signatures [Bol03a].

We present a transitive signature scheme $\mathcal{G}ap$-$\mathcal{TS}$ using these groups as well. It is proven transitively unforgeable under adaptive chosen-message attack assuming hardness of the one-more CDH problem as defined in Section 2.3.2.

This scheme is actually not of direct interest, because it is inferior to the $\mathcal{DL1m}$-$\mathcal{TS}$ scheme both with regard to assumptions made to prove security and with regard to performance. (In any group where one may implement $\mathcal{G}ap$-$\mathcal{TS}$, one may also implement $\mathcal{DL1m}$-$\mathcal{TS}$, and obtain security under weaker assumptions and with lower cost.) The value of $\mathcal{G}ap$-$\mathcal{TS}$ is that, unlike $\mathcal{DL1m}$-$\mathcal{TS}$ or $\mathcal{DL}$-$\mathcal{TS}$, it is amenable to the hash-based modification described next, resulting in $\mathcal{G}ap\mathcal{H}$-$\mathcal{TS}$, a scheme that has the shortest signatures amongst all schemes we

have discussed.

4.1.4 Eliminating Node Certificates via Hashing

THE $\mathcal{RSAH}$-$\mathcal{TS}$ SCHEME. The $\mathcal{RSA}$-$\mathcal{TS}$ scheme is amenable to a hash-based modification which eliminates the need for node certificates and thereby removes the standard signature scheme, and all its associated costs, from the picture. The public label of a node i is not chosen by the signer but rather implicitly specified as the output of a public hash function applied to i, and RSA decryption is used to compute edge labels. We prove that $\mathcal{RSAH}$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attack, assuming the hardness of one-more RSA-inversion in a model where the hash function is a random oracle.

THE $\mathcal{FactH}$-$\mathcal{TS}$ SCHEME. The fact that squaring modulo a composite is a trapdoor one-way function makes $\mathcal{Fact}$-$\mathcal{TS}$ amenable to a similar elimination of node certificates via hashing. We present the $\mathcal{FactH}$-$\mathcal{TS}$ transitive signature scheme where the public label of a node i is not chosen by the signer but rather specified via the output of a public hash function applied to i. (A difficulty, addressed in Section 4.6.1, is that the hash output might not be a quadratic residue.) We prove that $\mathcal{FactH}$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attacks in the random oracle model assuming factoring the underlying modulus is hard.

As Figure 4.1 indicates, the major cost savings is elimination of all costs associated to the standard scheme. However, signing now requires computation of square roots modulo N by the signer based on the prime factorization of N, which has cost comparable to an exponentiation modulo N. Thus overall the main gain is the reduction in signature size.

THE $\mathcal{GapH}$-$\mathcal{TS}$ SCHEME. The $\mathcal{Gap}$-$\mathcal{TS}$ scheme is also amenable to a similar hash-based modification, resulting in a scheme, $\mathcal{GapH}$-$\mathcal{TS}$, whose parameters are depicted in Figure 4.1. The signature here is simply a group element, and by the nature of Gap-DH groups, this means the $\mathcal{GapH}$-$\mathcal{TS}$ scheme has the shortest signatures of all.

$\mathcal{DL}$-$\mathcal{TS}$ AND $\mathcal{DL1m}$-$\mathcal{TS}$. The $\mathcal{DL}$-$\mathcal{TS}$ and $\mathcal{DL1m}$-$\mathcal{TS}$ schemes are not amenable to the hash-based modification since the discrete exponentiation function is not trapdoor over the groups used for these schemes.

STATEFUL VERSUS STATELESS SCHEMES. All of the five basic schemes $\mathcal{DL}$-$\mathcal{TS}$, $\mathcal{DL1m}$-$\mathcal{TS}$, $\mathcal{RSA}$-$\mathcal{TS}$, $\mathcal{Fact}$-$\mathcal{TS}$, $\mathcal{Gap}$-$\mathcal{TS}$ are stateful. As discussed in Section 4.3, there is a simple, general way to modify such schemes to obtain stateless ones. It may be interesting to note, however, that the $\mathcal{RSAH}$-$\mathcal{TS}$ and $\mathcal{GapH}$-$\mathcal{TS}$ schemes are naturally stateless. ($\mathcal{FactH}$-$\mathcal{TS}$ is not, and needs to be modified according to Section 4.3 if we want a stateless version.)

4.1.5 Definitional Contributions

Regarding the composability property, Micali and Rivest [MR02b, p. 238] (we have modified the notation to be consistent with ours) say: "... if someone sees Alice's signatures on edges $\{i, j\}$ and $\{j, k\}$ then that someone can easily compute a valid signature on edge $\{i, k\}$ that is indistinguishable from a signature on that edge that Alice would have produced herself." This seems to suggest that composition is only required to work when the given signatures were explicitly produced by the signer, but in fact we want composition to work even if the given signatures were themselves obtained via composition. Formulating an appropriate requirement turns out to be more delicate than one might imagine. One could require the simple condition that valid signatures (meaning, ones accepted by the verification algorithm relative to the signer's public key) can be composed to yield valid signatures. (This would follow Johnson et al. [JMSW02], who require a condition that implies this.) But this requirement is too strong in the current context. Indeed, as we show in Section 4.5, the $\mathcal{DL}$-$\mathcal{TS}$ scheme does not meet it, meaning there are valid signatures which, when composed, yield an invalid signature. The same is true for our schemes.

It can be proved that for $\mathcal{DL}$-$\mathcal{TS}$ and our schemes, finding valid signature inputs that make the composition algorithm return an invalid signature is computationally hard assuming the scheme is secure. But we prefer to not tie correctness of composition to security. Instead, we formulate correctness of composition via a recursive requirement that says that as long as one obtains signatures either directly via the signer or by applying the composition operation to signatures previously legitimately obtained or generated, then the resulting signature is valid. (This would be relatively easy to formulate if the signer was stateless, but needs more care due to the fact that the natural formulation of transitive signature schemes often results in a stateful signer.) As part of the formalization we provide in Definition 4.1, we follow Johnson et al. [JMSW02] and require a very strong form of the indistinguishability requirement mentioned in the quote above, namely that the signature output by the composition algorithm is not just indistinguishable from, but identical to, the one the signer would have produced. (As argued by Johnson et al. [JMSW02], this guarantees privacy.) The $\mathcal{DL}$-$\mathcal{TS}$ scheme, as well as all our schemes, meet this strong definition.

4.1.6 Related Work

Transitive signatures are one case of a more general concept promulgated by Rivest [Riv00] in talks, namely that of signature schemes that admit forgery of signatures derived by some specific operation on previous signatures but resist other forgeries. Johnson, Molnar, Song and Wagner [JMSW02] formalize a notion of homomorphic signature schemes that captures this. Context Extraction Signatures, as introduced earlier [SBZ02], as well as redactable signatures

and set-homomorphic signatures [JMSW02], fall in this framework. A signature scheme that is homomorphic with respect to the prefix operation is presented by Chari, Rabin and Rivest [CRR02].

The paper presented at the ASIACRYPT 2002 conference [BN00] contained the results pertaining to $\mathcal{RSA}$-$\mathcal{TS}$, and presented the new schemes $\mathcal{F}act$-$\mathcal{TS}$ and $\mathcal{F}act\mathcal{H}$-$\mathcal{TS}$. This thesis, besides including proofs omitted in the preliminary version, also adds the new schemes $\mathcal{DL}1m$-$\mathcal{TS}$, $\mathcal{G}ap$-$\mathcal{TS}$, $\mathcal{G}ap\mathcal{H}$-$\mathcal{TS}$.

There has been more work on transitive signatures subsequent to the appearance of our work [3]. Namely, Hohenberger [Hoh03] presents a general framework for the design and analysis of transitive signature schemes, as well as some results on the difficulty of constructing transitive signature schemes for directed graphs.

4.2 Definitions

4.2.1 Transitive Signature Schemes and their Correctness

All graphs in this chapter are undirected. If $G = (V, E)$ is a graph, its *transitive closure* is the graph $\widetilde{G} = (V, \widetilde{E})$ where $\{i, j\} \in \widetilde{E}$ iff there is a path from i to j in G. A graph $G^* = (V^*, E^*)$ is said to be *transitively closed* if for all nodes $i, j, k \in V^*$ such that $\{i, j\} \in E^*$ and $\{j, k\} \in E^*$, it also holds that $\{i, k\} \in E^*$; or in other words, edge $\{i, j\} \in E^*$ whenever there is a path from i to j in G^*. Note that the transitive closure of any graph G is a transitively closed graph. Also note that any transitively closed graph can be partitioned into connected components such that each component is a complete graph.

A *transitive signature (TS) scheme* $\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ is specified by four polynomial-time algorithms, and the functionality is as follows:

- The randomized *key generation* algorithm TKg takes as input 1^k, where $k \in \mathbb{N}$ is the security parameter, and returns a pair (tpk, tsk) consisting of a public key and matching secret key.
- The *signing algorithm* TSign, which could be stateful or randomized (or both), takes as input the secret key tsk and *nodes* $i, j \in \mathbb{N}$, and returns a value called an *original signature* of edge $\{i, j\}$ relative to tsk. If stateful, it maintains state which it updates upon each invocation.
- The deterministic *verification* algorithm TVf, given tpk, nodes $i, j \in \mathbb{N}$, and a candidate signature σ, returns either 1 or 0. In the former case we say that σ is a *valid* signature of edge $\{i, j\}$ relative to tpk.
- The deterministic *composition* algorithm Comp takes tpk, nodes $i, j, k \in \mathbb{N}$ and values σ_1, σ_2 to return either a value σ or a symbol $\perp$ to indicate failure.

$(tpk, tsk) \xleftarrow{R} \mathsf{TKg}(1^k)$
$S \leftarrow \emptyset$; $Legit \leftarrow \texttt{true}$; $NotOK \leftarrow \texttt{false}$
Run $\mathsf{F}(tpk : \text{TSign}, \text{Comp})$ replying to its oracle queries as follows:
  If F makes TSign query i, j then
    If $i = j$ then $Legit \leftarrow \texttt{false}$
    Else
      $\sigma \xleftarrow{R} \mathsf{TSign}(tsk, i, j)$; $S \leftarrow S \cup \{(\{i, j\}, \sigma)\}$
      If $\mathsf{TVf}(tpk, i, j, \sigma) = 0$ then $NotOK \leftarrow \texttt{true}$
  If F makes Comp query $i, j, k, \sigma_1, \sigma_2$ then
    If $[(\{i, j\}, \sigma_1) \notin S$ or $(\{j, k\}, \sigma_2) \notin S$ or i, j, k are not all distinct] then
      $Legit \leftarrow \texttt{false}$
    Else
      $\sigma \leftarrow \mathsf{Comp}(tpk, i, j, k, \sigma_1, \sigma_2)$; $S \leftarrow S \cup \{(\{i, k\}, \sigma)\}$
      $\tau \leftarrow \mathsf{TSign}(tsk, i, k)$
      If $[(\sigma \neq \tau)$ or $\mathsf{TVf}(tpk, i, k, \sigma) = 0]$ then $NotOK \leftarrow \texttt{true}$
When F halts, output $(Legit \wedge NotOK)$ and halt

Figure 4.3: Experiment used to define correctness of the transitive signature scheme $\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$.

The above formulation makes the simplifying assumption that the nodes of the graph are positive integers. In practice it is desirable to allow users to name nodes via whatever identifiers they choose, but these names can always be encoded as integers, so we keep the formulation simple.

Naturally, it is required that if σ is an original signature of edge $\{i, j\}$ relative to tsk then it is a valid signature of $\{i, j\}$ relative to tpk.

As discussed in Section 4.1.5, formulating a correctness requirement for the composition algorithm is more delicate. Micali and Rivest [MR02b] seem to suggest that composition is only required to work when the given signatures were explicitly produced by the signer, but in fact we want composition to work even if the given signatures were themselves obtained via composition. Furthermore the indistinguishability requirement is not formalized in [MR02b].

Definitions taking these issues into account are provided in the more general context of *homomorphic signature schemes* [JMSW02]. They ask that whenever the composition algorithm is invoked on valid signatures (valid meaning accepted by the verification algorithm relative to the signer's public key) it returns the same signature as the signer would produce. This captures indistinguishability in a strong way that guarantees privacy. However, one implication of their definition is that whenever the composition algorithm is invoked on valid signatures, it returns a valid signature, and this last property is not true

of known node certification based transitive signature schemes such as $\mathcal{DL}$-$\mathcal{TS}$, $\mathcal{RSA}$-$\mathcal{TS}$, and also not true for our new schemes. For all these schemes, it is possible to construct examples of valid signature inputs that, when provided to the composition algorithm, result in the latter failing (returning $\perp$ because it cannot compose) or returning an invalid signature, as we illustrate in Section 4.5. (Roughly, this happens because composition of a signature σ_1 of $\{i, j\}$ with a signature σ_2 of $\{j, k\}$ in these schemes requires that the public labels of node j as specified in σ_1 and σ_2 be the same. Validity of the individual signatures cannot guarantee this.)

This is not a weakness in the schemes, because in practice composition is applied not to arbitrary valid signatures but to ones that are legitimate, the latter being recursively defined: a signature is legitimate if it is either obtained directly by the signer, or obtained by applying the composition algorithm to legitimate signatures. What this points to is that we need to formulate a new correctness definition for composition that captures this intuition and results in a notion met by the known transitive signature schemes. Roughly, we would like a formulation that says that if the composition algorithm is invoked on legitimate signatures, then it returns the same signature as the signer would have produced. (Here, we are continuing to follow Johnson et al. [JMSW02] in capturing indistinguishability by the strong requirement that composed signatures are identical to original ones, but weakening their requirement by asking that this be true not for all valid signature inputs to the composition algorithm, but only for legitimate inputs.)

The formalization would be relatively simple (the informal description above would pretty much be it) if the signing algorithm were stateless, but the natural formulation of numerous transitive signature schemes seems to be in terms of a stateful signing algorithm. In this case, it is not clear what it means that the output of the composition algorithm is the same as that of the signer, since the latter's output depends on its internal state which could be different at different times. To obtain a formal definition of correctness that takes into account the statefulness of the signing algorithm, we proceed as follows. We associate to any algorithm F (deterministic, halting, but not computationally limited) and security parameter $k \in \mathbb{N}$ the experiment of Figure 4.3, which provides F with oracles

$$\textsc{TSign}(\cdot,\cdot) = \mathsf{TSign}(\mathit{tsk},\cdot,\cdot) \text{ and } \textsc{Comp}(\cdot,\cdot,\cdot,\cdot,\cdot) = \mathsf{Comp}(\mathit{tpk},\cdot,\cdot,\cdot,\cdot,\cdot)\ ,$$

where $\mathit{tpk}, \mathit{tsk}$ have been produced by running TKg on input 1^k. In this experiment, the TSign oracle maintains state, and updates this state each time it is invoked. It also tosses coins anew at each invocation if it is randomized.

Definition 4.1 We say that the transitive signature scheme $\mathcal{TS}$ is *correct* if for every (computationally unbounded) algorithm F and every k, the output of the experiment of Figure 4.3 is `true` with probability zero. ∎

The experiment computes a boolean *Legit* which is set to `false` if A ever makes an "illegitimate" query. It also computes a boolean *NotOK* which is set to `true` if a signature returned by the composition algorithm differs from the original one. To win, A must stay legitimate (meaning *Legit* = `true`) but violate correctness (meaning *NotOK* = `true`). The experiment returns `true` iff A wins. The definition requires that this happens with probability zero.

We say a transitive signature scheme is *non-trivial* if there is a polynomial p such that for all k, all tpk, tsk produced via TKg on input 1^k, and all $i, j \in \mathbb{N}$, if σ is a valid signature of edge $\{i, j\}$ relative to tpk, then the size of σ is at most $p(k)$. (This excludes schemes in which composition is performed by chaining.) We are only interested in non-trivial schemes, and all schemes in this chapter are non-trivial. We will not say this explicitly again.

4.2.2 Security of Transitive Signature Schemes

We recall the notion of security of transitive signature schemes [MR02b]. Associated to a transitive signature scheme $\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$, an adversary algorithm F and a security parameter $k \in \mathbb{N}$ is an experiment, denoted

$$\mathbf{Exp}^{\text{tu-cma}}_{\mathcal{TS},\mathsf{F}}(k) \ ,$$

that returns 1 if and only if F is successful in its attack on the scheme. The experiment begins by running TKg on input 1^k to get keys (tpk, tsk). It then runs F, providing this adversary with input tpk and access to an oracle $\textsc{TSign}(\cdot,\cdot) = \mathsf{TSign}(tsk, \cdot, \cdot)$. The oracle is assumed to maintain state or toss coins as needed. Let E be the set of all edges $\{i, j\}$ such that F made oracle query i, j, and let V be the set of all nodes involved in edges in E. Eventually, F will output $i', j' \in \mathbb{N}$ and a forgery σ'. We say that F *wins* the game if σ' is a valid signature of $\{i', j'\}$ relative to tpk but edge $\{i', j'\}$ is not in the transitive closure of graph $G = (V, E)$. The experiment returns 1 if F wins and 0 otherwise. The *advantage* of F in its attack on $\mathcal{TS}$ is the function $\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{TS},\mathsf{F}}(\cdot)$ defined for $k \in \mathbb{N}$ by

$$\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{TS},\mathsf{F}}(k) \ = \ \Pr\left[\, \mathbf{Exp}^{\text{tu-cma}}_{\mathcal{TS},\mathsf{F}}(k) = 1 \,\right] \ ,$$

where the probability is taken over all the random choices made in the experiment. We say that $\mathcal{TS}$ is *transitively unforgeable under adaptive chosen-message attack* if the function $\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{TS},\mathsf{F}}(\cdot)$ is negligible for any adversary F whose running time is polynomial in the security parameter k.

Some of our schemes are defined in the random oracle model (see Section 2.2), which means that the algorithms $\mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp}$ all have oracle access to one or more functions which in the correctness and security experiments are assumed to be drawn at random from appropriate spaces. Formally, both the experiment of Figure 4.3 and $\mathbf{Exp}^{\text{tu-cma}}_{\mathcal{TS},\mathsf{F}}(k)$ are augmented to choose a random function H mapping $\{0,1\}^*$ to an appropriate range, possibly depending on the public key,

and the adversary as well as the $\mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp}$ algorithms then get oracle access to this function. In Definition 4.1, the probability includes the choice of these functions, and so does the probability in the definition of $\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{TS},\mathsf{F}}(k)$.

4.3 Stateful versus Stateless Schemes

The signing algorithms of many transitive signature schemes are stateful. This is true for the $\mathcal{RSA}$-$\mathcal{TS}$ scheme, where it is important for composition that the signer associates a single public label to node i. As we will see, statefulness can also be important for security in that it associates to a public label $L(i)$ a single secret label $\ell(i)$. (The $\mathcal{Fact}$-$\mathcal{TS}$ and $\mathcal{FactH}$-$\mathcal{TS}$ schemes for example would otherwise soon give away two different square roots of $L(i)$, allowing an attacker to factor the modulus.) The $\mathcal{DL}$-$\mathcal{TS}$, $\mathcal{DL1m}$-$\mathcal{TS}$ and $\mathcal{Gap}$-$\mathcal{TS}$ schemes also have stateful signing algorithms.

In case one would like a stateless scheme, we note here a simple transformation that can be used to make the signer stateless, without loss of security or efficiency. Namely, let the signer's secret key include a key K specifying an instance F_K from a pseudorandom function family F [GGM86], and use $F_K(i)$ as the underlying coins (randomness) for all choices made by the signer related to node i. This enables the signer to recompute quantities as it needs them (rather than storing them), and yet be consistent, always creating the same quantities for a given node. In practice one can implement the pseudorandom function family via a block cipher. Since operation of a block cipher is significantly cheaper than the number-theoretic operations already being used in the transitive signature schemes, the stateless scheme will have a cost close to that of the original stateful one.

Having pointed this out, in the rest of the chapter we continue to work with stateful signing algorithms wherever they are more natural or convenient. We also note that, interestingly, the $\mathcal{RSAH}$-$\mathcal{TS}$ and $\mathcal{GapH}$-$\mathcal{TS}$ schemes are naturally stateless.

4.4 Transitive Signatures based on RSA

The $\mathcal{RSA}$-$\mathcal{TS}$ scheme was noted by Micali and Rivest [MR02b] as a simple alternative to $\mathcal{DL}$-$\mathcal{TS}$ which can be shown to be transitively unforgeable under non-adaptive chosen-message attacks assuming RSA is one-way. We do not know whether the same assumption suffices to prove it is transitively unforgeable under adaptive chosen-message attacks, but here we will show that this is true under a stronger assumption.

We defined an *RSA key generator* K_{rsa} in Section 2.3.1 as a randomized, polynomial-time algorithm that on input 1^k outputs a tuple (N, e, d) where

$ed \equiv 1 \mod \varphi(N)$. We do not attempt to pin down exactly how the generator operates, for example with regard to the distribution on the primes it chooses, or the choice of encryption exponent. (The latter may be chosen to be a small number, like 3, for efficiency, or a large number if desired.) All we ask is that the one-more RSA-inversion problem associated to the generator be hard. This makes our results more general.

THE SCHEME. We associate to any RSA key generator $\mathsf{K}_{\rm rsa}$ and any standard digital signature scheme $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ a transitive signature scheme $\mathcal{RSA}\text{-}\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ defined as follows:

- $\mathsf{TKg}(1^k)$ does the following

  (1.1) Run $\mathsf{SKg}(1^k)$ to generate a key pair (spk, ssk) for $\mathcal{SS}$
  (1.2) Run $\mathsf{K}_{\rm rsa}(1^k)$ to get a triple (N, e, d)
  (1.3) Output $tpk = (N, e, spk)$ as the public key and $tsk = (N, e, ssk)$ as the secret key.

  Note that the exponent d is discarded and in particular not part of the secret key.

- The signing algorithm TSign maintains state $St = (V, \ell, L, \Sigma)$ where $V \subseteq \mathbb{N}$ is the set of all queried nodes, the function $\ell\colon V \to \mathbb{Z}_N^*$ assigns to each node $i \in V$ a *secret label* $\ell(i) \in \mathbb{Z}_N^*$, while the function $L\colon V \to \mathbb{Z}_N^*$ assigns to each node $i \in V$ a *public label* $L(i)$, and the function $\Sigma\colon V \to \{0,1\}^*$ assigns to each node i a standard signature on $i \| L(i)$ under ssk. When invoked on inputs tsk, i, j, meaning when asked to produce a signature on edge $\{i, j\}$, it returns $\perp$ if $i = j$, and otherwise does the following:

  (2.1) If $i > j$ then $\mathrm{swap}(i, j)$
  (2.2) If $i \notin V$ then
  (2.3) $\quad V \leftarrow V \cup \{i\}$; $\ell(i) \xleftarrow{R} \mathbb{Z}_N^*$; $L(i) \leftarrow \ell(i)^e \bmod N$
  (2.4) $\quad \Sigma(i) \leftarrow \mathsf{SSign}(ssk, i \| L(i))$
  (2.5) If $j \notin V$ then
  (2.6) $\quad V \leftarrow V \cup \{j\}$; $\ell(j) \xleftarrow{R} \mathbb{Z}_N^*$; $L(j) \leftarrow \ell(j)^e \bmod N$
  (2.7) $\quad \Sigma(j) \leftarrow \mathsf{SSign}(ssk, j \| L(j))$
  (2.8) $\delta \leftarrow \ell(i)\ell(j)^{-1} \bmod N$
  (2.9) $C_i \leftarrow (i, L(i), \Sigma(i))$; $C_j \leftarrow (j, L(j), \Sigma(j))$
  (2.10) Return (C_i, C_j, δ) as the signature of $\{i, j\}$.

  We refer to $C_i = (i, L(i), \Sigma(i))$ as a *certificate* of node i.

- TVf, on input $tpk = (N, e, spk)$, nodes i, j and a candidate signature σ, proceeds as follows:

(3.1) If $i > j$ then swap(i, j)
(3.2) Parse σ as (C_i, C_j, δ), parse C_i as (i, L_i, Σ_i), parse C_j as (j, L_j, Σ_j)
(3.3) If $\mathsf{SVf}(spk, i\|L_i, \Sigma_i) = 0$ or $\mathsf{SVf}(spk, j\|L_j, \Sigma_j) = 0$ then return 0
(3.4) If $\delta^e \equiv L_i L_j^{-1} \bmod N$ then return 1 else return 0.

- The composition algorithm Comp takes tpk, nodes i, j, k and signatures σ_1 and σ_2, and computes a composed signature for edge $\{i, k\}$ as follows:

(4.1) If $i > k$ then swap(i, k) ; swap(σ_1, σ_2)
(4.2) Parse σ_1 as (C_1, C_2, δ_1) ; Parse σ_2 as (C_3, C_4, δ_2)
(4.3) If $i > j$ then swap(C_1, C_2) ; $\delta_1 \leftarrow \delta_1^{-1} \bmod N$
(4.4) If $j > k$ then swap(C_3, C_4) ; $\delta_2 \leftarrow \delta_2^{-1} \bmod N$
(4.5) $\delta \leftarrow \delta_1 \delta_2 \bmod N$
(4.6) Return (C_1, C_4, δ) as the signature for $\{i, k\}$.

The following proposition says that the $\mathcal{RSA}$-$\mathcal{TS}$ scheme meets our correctness definition for TS schemes. We note that it was to ensure that this correctness requirement is met that we have specified the composition algorithm above in full detail.

Proposition 4.2 The $\mathcal{RSA}$-$\mathcal{TS}$ transitive signature scheme described above satisfies the correctness requirement of Definition 4.1. ∎

We prove the above proposition using the two following claims. The first shows an invariant condition that holds at any time during the experiment, the second uses this invariant to show that the variable *NotOK* in the experiment of Figure 4.3 can never become `true`. From this, the above proposition follows.

Claim 4.3 If $St = (\ell, L, \Sigma, V)$ is the internal state of the TSign algorithm in $\mathcal{RSA}$-$\mathcal{TS}$, then at any time during the experiment in Figure 4.3, the following invariant holds true:

$$\mathit{Legit} = \texttt{false} \quad \vee \quad \forall\, (\{i, j\}, \sigma) \in S :$$

$$i \neq j \quad \wedge \quad \sigma = \begin{cases} \big((i, L(i), \Sigma(i)),\ (j, L(j), \Sigma(j)),\ \ell(i)\ell(j)^{-1} \bmod N\big) & \text{if } i < j \\ \big((j, L(j), \Sigma(j)),\ (i, L(i), \Sigma(i)),\ \ell(j)\ell(i)^{-1} \bmod N\big) & \text{if } j < i \end{cases} \tag{4.1}$$

Proof: We will prove the claim by induction on the number of TSign oracle queries q. Initially, $S = \emptyset$ and the claim is trivial. Suppose that the claim is true after $q - 1$ oracle queries. We will prove that it still holds after the qth oracle query.

If *Legit* = `false` before the qth query, then it will still be `false` after the qth query, directly proving the claim. We now concentrate on the case that *Legit* = `true`.

If the qth query is a TSIGN query i, j with $i = j$, *Legit* is set to `false`, again easily proving the claim. Otherwise, a new element $(\{i, j\}, \sigma)$ is added to S, where σ is the output of $\mathsf{TSign}(tsk, i, j)$. All elements of S that satisfied Equation (4.1) in the previous state of TSIGN, will still do so in the new state, because TSign only adds new entries to ℓ, L and Σ, but never changes existing entries. Therefore, it suffices to show that the newly added element $(\{i, j\}, \sigma)$ satisfies Equation (4.1). This can be seen from the description of the TSign algorithm. If $i < j$, it outputs a signature $\sigma = ((i, L(i), \Sigma(i)), (j, L(j), \Sigma(j)), \delta)$ with $\delta = \ell(i)\ell(j)^{-1} \bmod N$, as required. If $j < i$, TSign first swaps the values of i and j in line (2.1), such that the output of the algorithm is actually $\sigma = ((j, L(j), \Sigma(j)), (i, L(i), \Sigma(i)), \delta)$ with $\delta = \ell(j)\ell(i)^{-1} \bmod N$, again as required by Equation (4.1).

If the qth query is a COMP query $i, j, k, \sigma_1, \sigma_2$, we prove the claim as follows. If $(\{i, j\}, \sigma_1) \notin S$ or $(\{j, k\}, \sigma_2) \notin S$ or i, j, k are not all distinct, then *Legit* is set to `false` and the claim holds true. Otherwise, the composition algorithm is run to create $\sigma \leftarrow \mathsf{Comp}(tpk, i, j, k, \sigma_1, \sigma_2)$, and the element $(\{i, k\}, \sigma)$ is added to S. As the internal state of the TSIGN oracle is not affected by the composition algorithm, all elements that previously satisfied Equation (4.1) will still do so. We only have to check that the newly added element also satisfies Equation (4.1). If $i > k$ then i and k are swapped in line (4.1), as are the signatures σ_1 and σ_2. At this point we have signatures σ_1 and σ_2 for edges $\{i, j\}$ and $\{j, k\}$ satisfying equation Equation (4.1) with $i < k$. Let $\sigma_1 = (C_1, C_2, \delta_1)$, and let $\sigma_2 = (C_3, C_4, \delta_2)$. Line (4.3) of the Comp algorithm swaps C_1 and C_2 and inverts δ_1 if $i > j$, ensuring that after this step $C_1 = (i, L(i), \Sigma(i))$, $C_2 = (j, L(j), \Sigma(j))$ and $\delta_1 \equiv \ell(i)\ell(j)^{-1} \bmod N$. The same is done with C_3, C_4 and δ_2 if $j > k$ in line (4.4), ensuring that $C_3 = (j, L(j), \Sigma(j))$, $C_4 = (k, L(k), \Sigma(k))$ and $\delta_2 \equiv \ell(j)\ell(k)^{-1} \bmod N$. The signature that is finally returned is (C_1, C_4, δ), which indeed satisfies Equation (4.1) since $i < k$ and δ is computed as $\delta_1\delta_2 \equiv \ell(i)\ell(j)^{-1} \cdot \ell(j)\ell(k)^{-1} \equiv \ell(i)\ell(k)^{-1} \bmod N$. ∎

A corollary of the previous claim is that at any time during the experiment, $\mathsf{TVf}(tpk, i, j, \sigma) = 1$ for all $(\{i, j\}, \sigma) \in S$. From the description of the TSign algorithm, we can see that $L(i) \equiv \ell(i)^e \bmod N$ and $\Sigma(i)$ is a valid standard signature under *spk* for $i \| L(i)$. Given these facts and Equation (4.1), we can go through the description of TVf and check that it always returns 1.

Claim 4.4 The variable *NotOK* in the experiment in Figure 4.3 can never become `true`.

Proof: By the corollary above, the verification of a signature in S always succeeds, so the only way left for *NotOK* to become true during the experiment

is when $\sigma \neq \tau$ in a Comp query. The output of the signature algorithm for nodes i, k is $\tau = ((i, L(i), \Sigma(i)), (k, L(k), \Sigma(k)), \ell(i)\ell(k)^{-1})$ when $i < k$, and is $\tau = ((k, L(k), \Sigma(k)), (i, L(i), \Sigma(i)), \ell(k)\ell(i)^{-1})$ if $k < i$. We now prove that this is identical to the output of the composition algorithm when applied to nodes i, j, k and signatures σ_1, σ_2 such that $(\{i, j\}, \sigma_1), (\{j, k\}, \sigma_2) \in S$. In the proof of Claim 4.3, we already argued that the variables C_1 and C_4 by the end of the Comp algorithm are always assigned values $(i, L(i), \Sigma(i))$ and $(k, L(k), \Sigma(k))$, respectively, and that $\delta \equiv \ell(i)\ell(k)^{-1} \bmod N$. The values for i and k, however, might have been swapped in the first line of the Comp algorithm, so the returned signature is actually $\sigma = ((i, L(i), \Sigma(i)), (k, L(k), \Sigma(k)), \ell(i)\ell(k)^{-1})$ if $i < k$ and $\sigma = ((k, L(k), \Sigma(k)), (i, L(i), \Sigma(i)), \ell(k)\ell(i)^{-1})$ if $k < i$, exactly like τ. ∎

Since the experiment outputs $(\mathit{Legit} \wedge \mathit{NotOK})$ at the end of its execution, the previous claim implies that it returns `false` for every adversary A, thereby proving the correctness of $\mathcal{RSA}$-$\mathcal{TS}$.

Computational costs. As Figure 4.1 indicates, over and above costs associated to the standard signature scheme, signing and verifying require RSA encryptions, whose cost dominates that of quadratic-time operations such as multiplications and inverses mod N. The cost of the RSA encryptions is $O(|e| \cdot |N|^2)$ and depends on the choice of encryption exponent made by the RSA key generator; it can be small for a small exponent. Composition is efficient, involving only quadratic-time operations.

Security of $\mathcal{RSA}$-$\mathcal{TS}$. The following theorem says that as long as the RSA one-more-inversion problem is hard for the associated generator, and as long as the standard signature scheme is secure, the $\mathcal{RSA}$-$\mathcal{TS}$ transitive signature scheme is transitively unforgeable under adaptive chosen-message attack.

We first sketch the security proof of $\mathcal{RSA}$-$\mathcal{TS}$ against a non-adaptive adversary assuming the one-wayness of RSA and then explain why the same technique fails against adaptive adversaries. Subsequently we prove the scheme secure in the adaptive setting under the stronger one-more RSA assumption.

Given N, e, y and a non-adaptive adversary A attacking $\mathcal{RSA}$-$\mathcal{TS}$, we can compute x such that $x^e \equiv y \bmod N$ as follows. Generate a fresh key pair $(\mathit{spk}, \mathit{ssk})$ for $\mathcal{SS}$ and run A on $\mathit{tpk} = (N, e, \mathit{spk})$. The adversary A has to announce the entire set of edges E it wants to have signed before seeing any of the replies. If $G = (V, E)$ is the graph defined by the signature queries and $\widetilde{G} = (V, \widetilde{E})$ is its transitive closure, then $\widetilde{E}$ partitions V into $n \leq |V| \leq 2|E|$ disjoint components. Assuming the standard signature scheme is secure, the adversary cannot create its own node certificates, so a successful forgery must connect two nodes from different components in V. We sign all edges in E using the regular TSign algorithm, except for the edges in one randomly chosen 'special' component for which the public labels are computed as $L(i) \leftarrow \ell(i)^e \cdot y \bmod N$. Note that the validity of signatures is not affected by multiplying all public labels in a com-

ponent with the same factor. With probability of at least $2/n$, the adversary's forgery connects a node i in the special component to a node j in a different component.[1] Because of the assumed unforgeability of node certificates, the forgery provides us with a value δ such that $\delta^e \equiv L(i)L(j)^{-1} \equiv \ell(i)^e y \cdot \ell(j)^{-e} \bmod N$ and hence we can compute x as $\delta \cdot \ell(i)^{-1}\ell(j) \bmod N$.

The reason the above technique fails for adaptive adversaries is that there's no way to predict which nodes will be joined together in components. Not only would we have to guess one of the components that will be involved in the forgery, but also which nodes will be part of that component. The odds of guessing this correctly are in the order of $2^{-|E|}$, reducing our chance of inverting y to a negligible quantity. While the adaptive security of $\mathcal{RSA}$-$\mathcal{TS}$ under the one-wayness of RSA remains an open question, we found that it is provably secure under the stronger one-more RSA assumption, as stated in the following theorem.

Forging a signature for $\mathcal{RSA}$-$\mathcal{TS}$ is trivial if an insecure instance is used for the RSA key generator $\mathsf{K}_{\mathrm{rsa}}$ or the standard signature scheme $\mathcal{SS}$. The following theorem, however, states that the construction of $\mathcal{RSA}$-$\mathcal{TS}$ contains no weaknesses other than those induced by the underlying primitives.

Theorem 4.5 *Let $\mathsf{K}_{\mathrm{rsa}}$ be an RSA key generator and let $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ be a SS scheme. Let $\mathcal{RSA}$-$\mathcal{TS}$ be the TS scheme associated to $\mathsf{K}_{\mathrm{rsa}}$ and $\mathcal{SS}$ as defined above. If the one-more RSA problem associated to $\mathsf{K}_{\mathrm{rsa}}$ is hard and $\mathcal{SS}$ is unforgeable under adaptive chosen-message attack (*uf-cma *secure), then the $\mathcal{RSA}$-$\mathcal{TS}$ scheme is transitively unforgeable under adaptive chosen-message attack (*tu-cma *secure).* ∎

Proof: Suppose we are given a poly(k)-time adversary F for $\mathcal{RSA}$-$\mathcal{TS}$. We construct a one-more RSA adversary A, and a forger B attacking $\mathcal{SS}$, both poly(k)-time, such that for all k

$$\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{RSA}\text{-}\mathcal{TS},\mathsf{F}}(k) \leq \mathbf{Adv}^{\text{1m-rsa}}_{\mathsf{K}_{\mathrm{rsa}},\mathsf{A}}(k) + \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{B}}(k)\ . \tag{4.2}$$

The assumptions, namely that the one-more RSA problem associated to $\mathsf{K}_{\mathrm{rsa}}$ is hard and $\mathcal{SS}$ is unforgeable under adaptive chosen-message attack, imply that the advantage functions on the right-hand-side of Equation (4.2) are negligible. The equation then says that $\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{RSA}\text{-}\mathcal{TS},\mathsf{F}}(\cdot)$ is also negligible, which completes the proof. It remains to describe A and B.

The one-more RSA adversary A, as per the definitions above, gets inputs N, e and, has access to an inversion oracle $\textsc{Inv}(\cdot)$ and a challenge oracle $\textsc{Chall}$. It wins if it outputs the inverses of all points returned by $\textsc{Chall}$, using strictly less queries to the inversion oracle than it makes to the challenge oracle. Let us now

[1] We can even pump up this probability to 1/2 by choosing $n/2$ special components, but this is not crucial to the proof since all we need is non-negligible success probability.

describe how it operates. It first generates a fresh key pair (spk, ssk) for $\mathcal{SS}$ by running $\mathsf{SKg}(1^k)$. It then runs F on input $tpk = (N, e, spk)$. The idea is that when answering F's signature queries, A uses target points generated by the challenge oracle as public labels. Running the TSign algorithm in order to create signatures is not possible because A doesn't know the corresponding secret labels; instead, it sparingly uses the inversion oracle to compute edge labels, calling it only when the requested signature cannot be computed by composing previously signed edges. For this purpose, A maintains state information $St = (V, L, \Sigma, \Delta)$, where V, L and Σ are defined as in the TSign algorithm, but $\Delta : V \times V \to \mathbb{Z}_N^*$ is a function storing known edge labels. Now, in detail, when asked for a signature on edge $\{i, j\}$, A proceeds as follows:

(5.1) If $i > j$ then swap(i, j)
(5.2) If $i \notin V$ then
(5.3) $\quad V \leftarrow V \cup \{i\}$; $L(i) \leftarrow \textsc{Chall}(\varepsilon)$; $\Delta(i, i) \leftarrow 1$
(5.4) $\quad \Sigma(i) \leftarrow \mathsf{SSign}(ssk, i \| L(i))$
(5.5) If $j \notin V$ then
(5.6) $\quad V \leftarrow V \cup \{j\}$; $L(j) \leftarrow \textsc{Chall}(\varepsilon)$; $\Delta(j, j) \leftarrow 1$
(5.7) $\quad \Sigma(j) \leftarrow \mathsf{SSign}(ssk, j \| L(j))$
(5.8) If $\Delta(i, j)$ is not defined then
(5.9) $\quad \Delta(i, j) \leftarrow \textsc{Inv}(L(i) \cdot L(j)^{-1} \bmod N)$
(5.10) $\quad \Delta(j, i) \leftarrow \Delta(i, j)^{-1} \bmod N$
(5.11) $\quad$ For all $v \in V \setminus \{i, j\}$ do
(5.12) $\quad\quad$ If $\Delta(v, i)$ is defined then
(5.13) $\quad\quad\quad \Delta(v, j) \leftarrow \Delta(v, i) \cdot \Delta(i, j) \bmod N$
(5.14) $\quad\quad\quad \Delta(j, v) \leftarrow \Delta(v, j)^{-1} \bmod N$
(5.15) $\quad\quad$ If $\Delta(v, j)$ is defined then
(5.16) $\quad\quad\quad \Delta(v, i) \leftarrow \Delta(v, j) \cdot \Delta(j, i) \bmod N$
(5.17) $\quad\quad\quad \Delta(i, v) \leftarrow \Delta(v, i)^{-1} \bmod N$
(5.18) $\delta \leftarrow \Delta(i, j)$; Return $((i \| L(i), \Sigma(i)), (j \| L(j), \Sigma(j)), \delta)$ to F.

At the end of its execution, F outputs a forgery $\sigma' = ((i', L_{i'}, \Sigma_{i'}), (j', L_{j'}, \Sigma_{j'}), \delta')$ for edge $\{i', j'\}$. (During this analysis, we assume without loss of generality that $i' < j'$. If this is not the case, one can swap i' and j'.) Let $G = (V, E)$ be the graph defined by F's signature queries, and let $\widetilde{G} = (V, \widetilde{E})$ be its transitive closure. If σ' is not a valid forgery, meaning that $\mathsf{TVf}(tpk, i', j', \sigma') = 0$ or $\{i', j'\} \in \widetilde{E}$, then A aborts. Let $\mathbf{E}$ be the event that F's forgery contains recycled node certificates, i.e. $L_{i'} = L(i')$ and $L_{j'} = L(j')$. In case of the complementary event $\overline{\mathbf{E}}$, A aborts. Else it computes inverses of all challenges that it received from its challenge oracle, as follows. The transitively closed graph $\widetilde{G}$ is divided into c disjoint components $V_k \subset V$ for $k = 1 \ldots c$. Let $V_{k'}$ be the component containing node i'. For all $k = 1 \ldots c$, $k \neq k'$, algorithm A chooses a *reference node* $r_k \in V_k$ and computes the secret labels of all nodes in V_k as

(5.19) $\ell(r_k) \leftarrow \textsc{Inv}(L(r_k))$
(5.20) For all $v \in V_k \setminus \{r_k\}$ do
(5.21) $\quad \ell(v) \leftarrow \Delta(v, r_k) \cdot \ell(r_k) \bmod N$

while the secret labels of all nodes in component $V_{k'}$ are computed as

(5.22) $\ell(i') \leftarrow \delta' \cdot \ell(j') \bmod N$
(5.23) For all $v \in V_{k'} \setminus \{i'\}$ do
(5.24) $\quad \ell(v) \leftarrow \Delta(v, i') \cdot \ell(i')$.

From the way A answers F's signature queries, one can see that $\Delta(i, j)$ is defined for all nodes i, j that are in the same component, and hence that the values of Δ needed in the computations above are also defined. Algorithm A can now output the inverses of all its target points: for each $i \in V$, the public label $L(i)$ was obtained as a result of a query to $\textsc{Chall}()$, so the algorithm outputs $\ell(i)$ for all $i \in V$.

Now we need to check that A actually won the game. To do this we have to count the number of inversion queries. For each component V_k, $k \neq k'$, algorithm A needed $|V_k| - 1$ inversion queries to answer F's signature queries (the number of edges in a minimal spanning tree of V_k) plus one additional query at the end of the game to compute the secret label of r_k, summing up to $|V_k|$ inversion queries for each component. The component $V_{k'}$ only needed $|V_{k'}| - 1$ queries, because it did not need the additional query. So in summary, A inverted $|V|$ target points using $\sum_{k \neq k'} |V_k| + (|V_{k'}| - 1) = |V| - 1$ inversion queries, and hence wins the game.

The description of the forger B is rather straightforward: when run on input spk, it generates RSA parameters N, e, d using $\mathsf{K}_{\mathrm{rsa}}$ and runs F on input $tpk = (N, e, spk)$, answering its signature queries using the real TSign algorithm but consulting its own $\mathsf{SSign}(ssk, \cdot)$ oracle to create node certificates. In the event $\mathbf{E}$ that F's forgery recycles old node certificates, B gives up, but otherwise (in the event $\overline{\mathbf{E}}$) at least one of the node certificates contains a signature on a new message, and this can be used to output a forgery.

It is clear that A's simulation of F's environment is perfect. Accordingly we have

$$\begin{aligned}
\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{RSA}\text{-}\mathcal{TS},\mathsf{F}}(k) &= \Pr\left[\mathbf{Exp}^{\text{tu-cma}}_{\mathcal{RSA}\text{-}\mathcal{TS},\mathsf{F}}(k) = 1\right] \\
&= \Pr\left[\mathbf{Exp}^{\text{tu-cma}}_{\mathcal{RSA}\text{-}\mathcal{TS},\mathsf{F}}(k) = 1 \wedge \mathbf{E}\right] \\
&\quad + \Pr\left[\mathbf{Exp}^{\text{tu-cma}}_{\mathcal{RSA}\text{-}\mathcal{TS},\mathsf{F}}(k) = 1 \wedge \overline{\mathbf{E}}\right] \\
&\leq \mathbf{Adv}^{\text{1m-rsa}}_{\mathsf{K}_{\mathrm{rsa}},\mathsf{A}}(k) + \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{B}}(k) .
\end{aligned}$$

This yields Equation (4.2), as required. ∎

4.5 Definitional Issues with Correctness

We take a short side step to demonstrate the need for our (rather complicated) correctness definition. It is natural to consider the following alternative definition of correctness for composition. Say that a transitive signature scheme $\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ is *strongly-correct* if for every k, and every (tpk, tsk) that might be returned by $\mathsf{TKg}(1^k)$, we have:

If	σ_1 is a valid signature of edge $\{i, j\}$ relative to tpk, meaning that $\mathsf{TVf}(tpk, i, j, \sigma_1) = 1$
And	σ_2 is a valid signature of edge $\{j, k\}$ relative to tpk, meaning that $\mathsf{TVf}(tpk, j, k, \sigma_2) = 1$
And	$\sigma = \mathsf{Comp}(tpk, i, j, k, \sigma_1, \sigma_2)$
Then:	σ is a valid signature of edge $\{i, k\}$ relative to tpk, meaning that $\mathsf{TVf}(tpk, i, k, \sigma) = 1$

The purpose of this section is to point out that the $\mathcal{RSA}$-$\mathcal{TS}$ scheme does not meet this definition. Similar examples can be used to see that neither Micali and Rivest's $\mathcal{DL}$-$\mathcal{TS}$ scheme, nor any of our new schemes meet this definition. Note that the definition of Johnson et al. [JMSW02] implies strong-correctness as we have formulated it above (it requires more) and thus neither the $\mathcal{DL}$-$\mathcal{TS}$ scheme nor our schemes meet their definition.

We note that the "bad" inputs of our example can only be created by an adversary capable of forging standard signatures. However, we feel that composition is a "correctness" rather than a security requirement and should not rely on computational restrictions on adversaries.

We present a counter-example to show that the $\mathcal{RSA}$-$\mathcal{TS}$ scheme does not meet the above strong-correctness requirement. Suppose i, j, k are distinct nodes such that $i < j < k$. Suppose $\sigma_1 = (C_i, C_j, \delta_1)$ is a valid signature of $\{i, j\}$ relative to tpk, meaning

- $C_i = (i, L_i, \Sigma_i)$
- $C_j = (j, L_j, \Sigma_j)$
- $\delta_1^e \equiv L_i L_j^{-1} \bmod N$
- Σ_i a valid signature of $i \| L_i$ relative to spk
- Σ_j a valid signature of $j \| L_j$ relative to spk.

Also suppose $\sigma_2 = (C'_j, C_k, \delta_2)$ is a valid signature of $\{j, k\}$ relative to tpk, meaning that

- $C'_j = (j, L'_j, \Sigma'_j)$
- $C_k = (k, L_k, \Sigma_k)$

- $\delta_2^e \equiv L'_j L_k^{-1} \bmod N$
- Σ'_j a valid signature of $j \| L'_j$ relative to *spk*
- Σ_k a valid signature of $k \| L_k$ relative to *spk*.

On inputs $tpk, i, j, k, \sigma_1, \sigma_2$, the composition algorithm of the $\mathcal{RSA}$-$\mathcal{TS}$ scheme is defined to return (C_i, C_k, δ_3) where

$$\delta_3 \equiv \delta_1 \cdot \delta_2 \bmod N \ .$$

Now, using the above, we have

$$\begin{aligned} \delta_3^e &\equiv \delta_1^e \cdot \delta_2^e \\ &\equiv L_i L_j^{-1} \cdot L'_j L_k^{-1} \bmod N \ . \end{aligned}$$

For the verification algorithm to accept, the above should equal $L_i L_k^{-1} \bmod N$, meaning the verification algorithm would only accept (C_i, C_k, δ_3) as a valid signature of $\{i, k\}$ relative to *tpk* if $L_j = L'_j$. However, the validity of the given signatures σ_1, σ_2 does not imply that $L_j = L'_j$. Accordingly, we have an example of valid signatures yielding, via composition, an invalid signature. This shows that the $\mathcal{RSA}$-$\mathcal{TS}$ scheme is not strongly-correct.

Note that creating the valid signatures σ_1, σ_2 that make composition fail, even given oracle access to the TSign algorithm, would require forging relative to the standard scheme, so in practice we do not expect the composition algorithm to receive these inputs. However, it is inconvenient to formulate a correctness requirement that hinges on security.

As we have shown in Proposition 4.2, the $\mathcal{RSA}$-$\mathcal{TS}$ scheme is, however, correct as per Definition 4.1. Even though an algorithm A in the experiment of Figure 4.3 is not computationally restricted and could create σ_1, σ_2 as above and invoke the composition algorithm, examination of the experiment shows that the flag *Legit* would be set to `false`, and thus A would not win, so our definition is not violated.

4.6 New Schemes

We describe three new transitive signature schemes, all proven transitively unforgeable under adaptive chosen-message attack.

4.6.1 The $\mathcal{F}act$-$\mathcal{TS}$ Scheme

Our factoring-based transitive signature ($\mathcal{F}act$-$\mathcal{TS}$) scheme stays within the node certification technique but, by implementing label algebra via square roots modulo a composite, provides security based on factoring while reducing some costs compared to $\mathcal{DL}$-$\mathcal{TS}$ and $\mathcal{RSA}$-$\mathcal{TS}$.

The scheme is parameterized with a modulus generator K_{fact} as defined in Section 2.3.1. There are numerous possible modulus generators which differ in the structure of the primes chosen or the distribution under which they are chosen. We do not restrict the type of generator, but only assume that the associated factoring problem is hard.

THE SCHEME. We associate to any modulus generator K_{fact} and any standard digital signature scheme $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ a transitive signature scheme $\mathcal{Fact}\text{-}\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ defined as follows:

- $\mathsf{TKg}(1^k)$ proceeds as in $\mathcal{RSA}\text{-}\mathcal{TS}$ with the following changes:

  (1.2) Run $\mathsf{K}_{\text{fact}}(1^k)$ to get a triple (N, p, q)
  (1.3) Output $tpk = (N, spk)$ as the public key and $tsk = (N, ssk)$ as the secret key.

  Note that the primes p, q are discarded and in particular are not part of the secret key.

- The signing algorithm TSign maintains state as in $\mathcal{RSA}\text{-}\mathcal{TS}$. On inputs tsk, i, j it proceeds as the TSign algorithm of $\mathcal{RSA}\text{-}\mathcal{TS}$ except that rather than computing $L(\cdot)$ as $\ell(\cdot)^e$ it computes $L(\cdot)$ as $\ell(\cdot)^2$ in lines (2.3) and (2.6).

- TVf, on input $tpk = (N, spk)$, nodes i, j and a candidate signature σ, proceeds exactly as the TVf algorithm of the $\mathcal{RSA}\text{-}\mathcal{TS}$ scheme, except that δ^e is replaced with δ^2 in line (3.4).

- The composition algorithm is identical to that of $\mathcal{RSA}\text{-}\mathcal{TS}$.

A proof by induction similar to the one of Proposition 4.2 can be used to show the $\mathcal{Fact}\text{-}\mathcal{TS}$ transitive signature scheme described above satisfies the correctness requirement of Definition 4.1.

COMPUTATIONAL COSTS. The cost for the signature algorithm is dominated by multiplications and inversions modulo N, for both of which there exist algorithms quadratic in $|N|$, and the cost of generating two standard signatures, which depends on the choice of underlying standard signature scheme. It is not strictly necessary to test membership in $\mathbb{Z}_N^*$, because it is very unlikely that a randomly generated value is not coprime with N. (Otherwise N could be easily factored by computing the greatest common divisor with a random integer.) Verification takes a couple of multiplications mod N and two standard signature verifications. The composition of two signatures involves one multiplication and possibly an inversion in $\mathbb{Z}_N^*$. See Figure 4.1 for the cost summary.

SECURITY. The following is the formal statement of our result about the security of $\mathcal{Fact}\text{-}\mathcal{TS}$.

Theorem 4.6 *Let* K_{fact} *be a modulus generator and let* $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ *be a SS scheme. Let* $\mathcal{Fact}$-$\mathcal{TS}$ *be the TS scheme associated to them as defined above. If the factoring problem associated to* K_{fact} *is hard and* $\mathcal{SS}$ *is unforgeable under adaptive chosen-message attack, then* $\mathcal{Fact}$-$\mathcal{TS}$ *is transitively unforgeable under adaptive chosen-message attack.* ∎

Although $\mathcal{RSA}$-$\mathcal{TS}$ and $\mathcal{Fact}$-$\mathcal{TS}$ are very similar, the security of the latter is based on a weaker assumption. Intuitively, the reason is that RSA induces a permutation on $\mathbb{Z}_N^*$, whereas squaring maps 4 different elements of $\mathbb{Z}_N^*$ to the same square and two different roots of the same square reveal the factorization of the modulus.

Proof of Theorem 4.6. Suppose we are given a poly(k)-time adversary F for $\mathcal{Fact}$-$\mathcal{TS}$. We construct a factoring adversary A, and a forger B attacking $\mathcal{SS}$, both poly(k)-time, such that for all k

$$\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{Fact}\text{-}\mathcal{TS},\mathsf{F}}(k) \leq 2 \cdot \mathbf{Adv}^{\text{fact}}_{\mathsf{K}_{\text{fact}},\mathsf{A}}(k) + \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{B}}(k)\ . \tag{4.3}$$

The assumptions made in the theorem conclude the proof. It remains to describe A and B.

The factoring algorithm A gets as input a modulus N generated by K_{fact} and begins by picking keys for the standard signature scheme via $(spk, ssk) \xleftarrow{R} \mathsf{SKg}(1^k)$. It then lets $tpk = (N, spk)$ be a public key for the transitive signature scheme and starts running F on input tpk. To reply to F's TSign oracle queries, algorithm A simply runs the TSign procedure of the transitive signature scheme, which it can because it possesses the secret key $tsk = (N, ssk)$ corresponding to tpk. (We use here the fact that signing does not require knowledge of the prime factors of N.) A maintains the state information $St = (V, \ell, L, \Sigma)$ of the TSign procedure. Once F is done querying its oracle, it will output its forgery $\sigma' = ((i', L_{i'}, \Sigma_{i'}), (j', L_{j'}, \Sigma_{j'}), \delta')$ for edge $\{i', j'\}$. (We again assume without loss of generality that $i' < j'$.) Let $G = (V, E)$ be the graph defined by F's signature queries, and let $\widetilde{G} = (V, \widetilde{E})$ denote its transitive closure. Let $\mathbf{E}_1$ denote the event that σ' is a certificate-recycling forgery (i.e. $L_{i'} = L(i')$ and $L_{j'} = L(j')$), and let $\mathbf{E}_2$ be the event that $\delta' \equiv \pm\delta$ where $\delta \equiv \ell(i')\ell(j')^{-1} \bmod N$. If σ' is not a valid forgery, or in the event that $\overline{\mathbf{E}}_1 \vee \mathbf{E}_2$, algorithm A gives up. Otherwise, it computes and returns $r = \gcd(\delta + \delta', N)$, which is a factor of N because δ and δ' are different square roots of $L(i')L(j')^{-1} \bmod N$. This completes the description of factoring algorithm A.

With regard to the analysis, it is tempting to say that since $\ell(i')$ and $\ell(j')$ were chosen at random, with probability 1/2 A now has two square roots δ and δ' such that $\delta \not\equiv \pm\delta' \bmod N$, enabling it to factor N. This argument would be correct if the forger were only given $L(i')$ and $L(j')$, without having any further information on exactly which root A knows. However, by signing edges involving

nodes i' or j', algorithm A might have given away some additional information about its choices for $\ell(i')$ and $\ell(j')$. It is crucial to the security of the scheme that this information doesn't help the forger in creating a forgery with edge label $\delta' \equiv \pm\delta$, as this would annihilate A's advantage in factoring N. Fortunately, it turns out that the exact value of δ remains information-theoretically hidden from the forger as long as $\{i', j'\}$ is not in the transitive closure of the signed edges. The crucial fact, which we will justify later, is that

$$\Pr\left[\overline{\mathbf{E}_2} \mid \mathbf{E}_1 \wedge \mathbf{Exp}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) = 1\right] \;=\; \frac{1}{2}\,. \tag{4.4}$$

Given this, we have

$$\begin{aligned}
\mathbf{Adv}^{\text{fact}}_{\mathsf{K}_{\text{fact}},\mathsf{A}}(k) &\geq \Pr\left[\mathbf{Exp}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) = 1 \wedge \mathbf{E}_1 \wedge \overline{\mathbf{E}_2}\right] \\
&= \Pr\left[\overline{\mathbf{E}_2} \mid \mathbf{E}_1 \wedge \mathbf{Exp}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) = 1\right] \\
&\quad \cdot \Pr\left[\mathbf{E}_1 \wedge \mathbf{Exp}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) = 1\right] \\
&= \frac{1}{2} \cdot \Pr\left[\mathbf{E}_1 \wedge \mathbf{Exp}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) = 1\right]\,.
\end{aligned}$$

The description of algorithm B breaking $\mathcal{SS}$ is very similar to the description of algorithm B in the proof of Theorem 4.5. Details are omitted. As in that proof we will have

$$\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{B}}(k) \geq \Pr\left[\overline{\mathbf{E}}_1 \wedge \mathbf{Exp}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) = 1\right]\,.$$

Putting the above together we have

$$\begin{aligned}
\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) &= \Pr\left[\mathbf{E}_1 \wedge \mathbf{Exp}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) = 1\right] \\
&\quad + \Pr\left[\overline{\mathbf{E}}_1 \wedge \mathbf{Exp}^{\text{tu-cma}}_{\mathcal{F}act\text{-}\mathcal{TS},\mathsf{F}}(k) = 1\right] \\
&\leq 2 \cdot \mathbf{Adv}^{\text{fact}}_{\mathsf{K}_{\text{fact}},\mathsf{A}}(k) + \mathbf{Adv}^{\text{uf-cma}}_{\mathcal{SS},\mathsf{B}}(k)
\end{aligned}$$

as desired. It remains to justify Equation (4.4).

Let $G = (V, E)$ be the graph defined by the forger's signature queries, and let $\widetilde{G} = (V, \widetilde{E})$ be the transitive closure of G. We represent A's secret information by a random variable $\boldsymbol{\ell}$ that is distributed uniformly over *Secrets*, the set of all functions from V to $\mathbb{Z}_N^*$. The forger's view consists of a function L assigning a square modulo N to each node in V, and a function Δ assigning an edge label in $\mathbb{Z}_N^*$ to each edge in $\widetilde{E}$. (We ignore the standard digital signatures on the node certificates, as they are irrelevant for this analysis.) However, not just any pair of functions $\langle L, \Delta\rangle$ can occur as the forger's view. We say that forger view

$\langle L, \Delta \rangle$ is *consistent* with $\ell \in \mathit{Secrets}$ (and vice versa that ℓ is consistent with $\langle L, \Delta \rangle$) if and only if

$$L(i) \equiv \ell(i)^2 \bmod N \quad \text{for all } i \in V \tag{4.5}$$

$$\Delta(i,j) \equiv \ell(i)\ell(j)^{-1} \bmod N \quad \text{for all } \{i,j\} \in \widetilde{E}\ ,\ i < j \tag{4.6}$$

The set of all possible forger views *Views* can then be defined as the set of all pairs $\langle L, \Delta \rangle$ that are consistent with some $\ell \in \mathit{Secrets}$. The actual view of the forger is a random variable **View** distributed over *Views* as induced by $\boldsymbol{\ell}$. The following lemma states that for every $\langle L, \Delta \rangle \in \mathit{Views}$ and for every $\{i', j'\} \notin \widetilde{E}$, any square root δ of $L(i')L(j')^{-1} \bmod N$ is equally likely to be $\boldsymbol{\delta} \equiv \boldsymbol{\ell}(i')\boldsymbol{\ell}(j')^{-1} \bmod N$, the root A has "in mind", when given only $\mathbf{View} = \langle L, \Delta \rangle$, and hence that no forger, on input only **View**, can predict $\boldsymbol{\delta}$ with higher probability of success than random guessing.

Equation (4.4) follows easily from the following lemma. Its proof completes the proof of Theorem 4.6.

Lemma 4.7 For any $\langle L, \Delta \rangle \in \mathit{Views}$, for any $\{i', j'\} \notin \widetilde{E}$ and for any $\delta \in \mathbb{Z}_N^*$ with $\delta^2 \equiv L(i')L(j')^{-1} \bmod N$:

$$\Pr\left[\boldsymbol{\delta} \equiv \delta \bmod N \mid \mathbf{View} = \langle L, \Delta \rangle\right] = \frac{1}{4}\ . \tag{4.7}$$

∎

Proof: Since the outcome of all random variables is uniquely determined by the signer's choice for $\boldsymbol{\ell}$, we can reduce all probabilities on random variables to the probability of making some particular choice for $\boldsymbol{\ell}$. For example, if we define $\mathit{Cons}(\langle L, \Delta \rangle) \subseteq \mathit{Secrets}$ to be the set of all $\ell \in \mathit{Secrets}$ consistent with $\langle L, \Delta \rangle$, then we can replace $\Pr[\mathbf{View} = \langle L, \Delta \rangle]$ with $\Pr[\boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle)]$. Using this fact and some basic probability theory, we can write

$$\begin{aligned}
&\Pr\left[\boldsymbol{\delta} = \delta \mid \mathbf{View} = \langle L, \Delta \rangle\right] \\
&\quad = \Pr\left[\boldsymbol{\delta} = \delta \ \wedge\ \mathbf{View} = \langle L, \Delta \rangle \mid \mathbf{View} = \langle L, \Delta \rangle\right] \\
&\quad = \Pr\left[\boldsymbol{\delta} = \delta \ \wedge\ \boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle) \mid \boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle)\right] \\
&\quad = \Pr\left[\boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle) \mid \boldsymbol{\delta} = \delta \ \wedge\ \boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle)\right] \\
&\qquad \cdot \frac{\Pr\left[\boldsymbol{\delta} = \delta \ \wedge\ \boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle)\right]}{\Pr\left[\boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle)\right]} \\
&\quad = 1 \cdot \frac{\Pr\left[\boldsymbol{\delta} = \delta \ \wedge\ \boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle)\right]}{\Pr\left[\boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta \rangle)\right]}\ .
\end{aligned} \tag{4.8}$$

We want to find a numerical expression for the last two factors in Equation (4.8). Because $\boldsymbol{\ell}$ is uniformly distributed over *Secrets*, the probability that $\boldsymbol{\ell} \in S \subseteq$ *Secrets* is simply the number of elements in S divided by $|\mathit{Secrets}| = \varphi(N)^{|V|}$.

We first try to find an expression for the number of elements in $\mathit{Cons}(\langle L, \Delta\rangle)$. For ℓ to be consistent with forger view $\langle L, \Delta\rangle$, it has to satisfy the system of equations given by (4.5) and (4.6). Considering only equations (4.5), there are four possibilities for $\ell(i)$ left for every $i \in V$, namely the four square roots of $L(i)$. Equations (4.6) impose additional restrictions on ℓ. Many of these are linearly dependent, though. In order to count the actual number of possible solutions, we'd like to replace (4.6) with an equivalent but linearly independent set of equations.

Let c be the number of disjoint components $V_k \subset V$ in the transitively closed graph $\widetilde{G} = (V, \widetilde{E})$. If we define one node r_k in each component V_k to be the *reference node* for that component, and denote the reference node in the component of node i as $R(i)$, then the equations in the following system are clearly linearly independent:

$$\Delta(i, R(i)) \equiv \ell(i)\ell(R(i))^{-1} \bmod N \quad \text{for all } i \in V \setminus \{r_k \mid k = 1 \ldots c\}. \tag{4.9}$$

At the same time, they also form a system equivalent to (4.6), because every equation in (4.6) is either contained in (4.9), or can be written as the quotient of two equations in (4.9). The equations in (4.9) imply that once ℓ is fixed for the c reference nodes, ℓ is completely defined. Together with Equation (4.5), that leaves c entries of ℓ to be chosen freely from four values, so

$$\Pr\left[\boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta\rangle)\right] = \frac{4^c}{\varphi(N)^{|V|}} \ . \tag{4.10}$$

To what amount does the addition of the requirement $\boldsymbol{\delta} = \delta$ restrict our choices for $\boldsymbol{\ell}$? This comes down to adding

$$\ell(i')\ell(j')^{-1} \equiv \delta \bmod N$$

to the systems given by (4.5) and (4.9), or equivalently, adding the equation

$$\ell(R(i')) \equiv \delta \cdot \Delta(i', R(i'))^{-1} \cdot \Delta(j', R(j')) \cdot \ell(R(j')) \bmod N$$

which directly links $\ell(R(i'))$ to the choice for $\ell(R(j'))$. So now there are only $c - 1$ entries of ℓ left to choose, giving

$$\Pr\left[\boldsymbol{\delta} = \delta \ \wedge\ \boldsymbol{\ell} \in \mathit{Cons}(\langle L, \Delta\rangle)\right] = \frac{4^{c-1}}{\varphi(N)^{|V|}} \ . \tag{4.11}$$

Substituting the factors in Equation (4.8) with Equations (4.10) and (4.11) yields Equation (4.7), thereby proving the lemma. ∎

4.6.2 The $\mathcal{DL}1m$-$\mathcal{TS}$ Scheme

Micali and Rivest's $\mathcal{DL}$-$\mathcal{TS}$ scheme [MR02b] uses two generators, which is important to their security proof. The underlying ideas trace back to Okamoto's two-generator-based $\mathcal{OKCL}$-$\mathcal{SI}$ scheme [Oka93] (see Figure 3.18) and its proof of security against impersonation under active attack. However, Schnorr's *Schnorr-SI* scheme [Sch90] (see Figure 3.21), which uses only a single generator, is simpler, more natural and has lower cost, particularly in size of secret keys. We ask whether there is an analogous single-generator-based transitive signature scheme. We answer this in the affirmative, presenting $\mathcal{DL}1m$-$\mathcal{TS}$, which is a simpler and somewhat more natural single-generator based version of $\mathcal{DL}$-$\mathcal{TS}$, offering some performance improvements. However, while the proof of security of $\mathcal{DL}$-$\mathcal{TS}$ relied only on the standard hardness of discrete logarithms assumption, the proof of security of $\mathcal{DL}1m$-$\mathcal{TS}$ relies on the hardness of the one-more discrete logarithm problem (see Section 2.3.2). This is again analogous to the situation for identification schemes. Okamoto proved his scheme secure under the standard hardness of discrete logarithm assumption, while the proof of security of Schnorr's scheme (which remained an open problem for a while) is based on the hardness of the one-more discrete logarithm problem [BP02]. The $\mathcal{DL}1m$-$\mathcal{TS}$ scheme is similar to $\mathcal{RSA}$-$\mathcal{TS}$.

In Section 2.3.2, we introduced discrete logarithm group generators K_{dlog} as randomized polynomial-time algorithms that on input 1^k output a tuple $(\mathbb{G}, g, q)$, where q is an odd prime, $\mathbb{G}$ is (the compact description of) a cyclic group of order q, and g is a generator of $\mathbb{G}$. We do not attempt to pin down exactly how the generator operates. In particular there are many classes of groups (of prime order) which could be used. One example is that $2q+1$ is a prime and $\mathbb{G}$ is the subgroup of quadratic residues of $\mathbb{Z}^*_{2q+1}$. Another possibility is elliptic curve groups. This makes our results more general.

THE SCHEME. We associate to any cyclic group generator K_{dlog} and any standard digital signature scheme $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ a transitive signature scheme $\mathcal{DL}1m$-$\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ defined as follows:

- $\mathsf{TKg}(1^k)$ proceeds as in $\mathcal{RSA}$-$\mathcal{TS}$ with the following changes:

  (1.2) Run $\mathsf{K}_{\text{dlog}}(1^k)$ to get a triple $(\mathbb{G}, g, q)$
  (1.3) Output $tpk = (\mathbb{G}, g, q, spk)$ as the public key and
  (1.4) $tsk = (\mathbb{G}, g, q, ssk)$ as the secret key.

- The signing algorithm TSign maintains state $St = (V, \ell, L, \Sigma)$ where V, Σ are as in $\mathcal{RSA}$-$\mathcal{TS}$, ℓ: $V \to Z_q$ and L: $V \to \mathbb{G}$. When invoked on inputs tsk, i, j, it proceeds as the TSign algorithm of $\mathcal{RSA}$-$\mathcal{TS}$ except for the following changes:

(2.2) If $i \notin V$ then
(2.3) $\quad V \leftarrow V \cup \{i\}$; $\ell(i) \xleftarrow{R} \mathbb{Z}_q$; $L(i) \leftarrow g^{\ell(i)}$
(2.5) If $j \notin V$ then
(2.6) $\quad V \leftarrow V \cup \{j\}$; $\ell(j) \xleftarrow{R} \mathbb{Z}_q$; $L(j) \leftarrow g^{\ell(j)}$
(2.8) $\delta \leftarrow \ell(i) - \ell(j) \bmod q$.

- TVf, on input $tpk = (\mathbb{G}, g, q, spk)$, nodes i, j and a candidate signature σ, proceeds as the TVf algorithm of the $\mathcal{RSA}$-$\mathcal{TS}$ scheme, except for the following change:

  (3.4) If $g^{\delta} \equiv L_i L_j^{-1}$ then return 1 else return 0.

- The composition algorithm Comp takes tpk, nodes i, j, k and signatures σ_1 and σ_2, and proceeds as the Comp algorithm of $\mathcal{RSA}$-$\mathcal{TS}$ except for the following changes. In line (4.3) the computation of δ_1^{-1} is in $\mathbb{G}$ rather than being modulo N, and similarly for line (4.4). Also line (4.5) is replaced with

  (4.5) $\delta \leftarrow \delta_1 + \delta_2 \bmod q$.

This scheme offers some performance benefits compared to $\mathcal{DL}$-$\mathcal{TS}$, as indicated in Figure 4.1, namely a reduced signature size and composition time.

As the above indicates, this scheme is very similar to $\mathcal{RSA}$-$\mathcal{TS}$, replacing "$x^e \bmod N$" (with $x \in \mathbb{Z}_N^*$) by "g^x" (with $x \in \mathbb{Z}_q$). Accordingly a proof similar to the one of Proposition 4.2 can be used to show that $\mathcal{DL}1m$-$\mathcal{TS}$ satisfies the correctness requirement of Definition 4.1, and the following security result can be established by a proof analogous to that of Theorem 4.5, with the role of Inv played by Dlog.

Theorem 4.8 *Let $\mathsf{K}_{\mathrm{dlog}}$ be a discrete logarithm group generator and let $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ be a SS scheme. Let $\mathcal{DL}1m$-$\mathcal{TS}$ be the TS scheme associated to them as defined above. If the one-more discrete logarithm problem associated to $\mathsf{K}_{\mathrm{dlog}}$ is hard and $\mathcal{SS}$ is unforgeable under adaptive chosen-message attack, then $\mathcal{DL}1m$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attack.* ∎

4.6.3 The $\mathcal{Gap}$-$\mathcal{TS}$ Scheme

As indicated in Section 4.1.3, the $\mathcal{Gap}$-$\mathcal{TS}$ scheme is inferior to the $\mathcal{DL}1m$-$\mathcal{TS}$ scheme, requiring stronger assumptions and yet providing poorer performance. We describe it because, unlike $\mathcal{DL}1m$-$\mathcal{TS}$, it is amenable to the hash-based modification we describe later, leading to the scheme having the shortest signatures amongst all the schemes we have discussed.

As introduced in Section 2.3.2, Gap Diffie-Hellman groups are cyclic groups in which the decisional Diffie-Hellman (DDH) problem is efficiently solved by

the $\mathsf{S}_{\mathrm{ddh}}$ algorithm, while the computational Diffie-Hellman (CDH) problem is hard.

THE SCHEME. We associate to any Gap-DH group specifier $(\mathsf{K}_{\mathrm{gap}}, \mathsf{S}_{\mathrm{ddh}})$ and any standard digital signature scheme $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ a transitive signature scheme $\mathcal{Gap\text{-}TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ defined as follows:

- $\mathsf{TKg}(1^k)$ proceeds as in $\mathcal{RSA\text{-}TS}$ with the following changes:

  (1.2) $(\hat{\mathbb{G}}, g, q) \xleftarrow{R} \mathsf{K}_{\mathrm{gap}}(1^k)$; $a \xleftarrow{R} \mathbb{Z}_q$; $u \leftarrow g^a$
  (1.3) Output $tpk = (\hat{\mathbb{G}}, q, g, u, spk)$ and $tsk = (\hat{\mathbb{G}}, q, g, u, ssk)$.

  Note that a is discarded.

- The signing algorithm TSign maintains state $St = (V, \ell, L, \Sigma)$ where V, Σ are as in $\mathcal{RSA\text{-}TS}$ and ℓ, L: $V \to \hat{\mathbb{G}}$. When invoked on inputs tsk, i, j, it proceeds as the TSign algorithm of $\mathcal{RSA\text{-}TS}$ except for the following changes:

  (2.2) If $i \notin V$ then
  (2.3) $\quad V \leftarrow V \cup \{i\}$; $b_i \xleftarrow{R} \mathbb{Z}_q$; $\ell(i) \leftarrow u^{b_i}$; $L(i) \leftarrow g^{b_i}$
  (2.5) If $i \notin V$ then
  (2.6) $\quad V \leftarrow V \cup \{j\}$; $b_j \xleftarrow{R} \mathbb{Z}_q$; $\ell(j) \leftarrow u^{b_j}$; $L(j) \leftarrow g^{b_j}$
  (2.8) $\delta \leftarrow \ell(i)\ell(j)^{-1}$.

- TVf, on input $tpk = (\hat{\mathbb{G}}, q, g, u, spk)$, nodes i, j and a candidate signature σ, proceeds as the TVf algorithm of the $\mathcal{RSA\text{-}TS}$ scheme, except for the following change:

  (3.4) If $\mathsf{S}_{\mathrm{ddh}}(\hat{\mathbb{G}}, q, g, u, L_i L_j^{-1}, \delta) = 1$ then return 1 else return 0.

  That is, it checks that δ is the solution to the CDH problem $(g, u, L_i L_j^{-1})$.

- The composition algorithm Comp takes tpk, nodes i, j, k and signatures σ_1 and σ_2, and proceeds as the Comp algorithm of $\mathcal{RSA\text{-}TS}$ except for the following changes. In line (4.3) the computation of δ_1^{-1} is in $\hat{\mathbb{G}}$ rather than being modulo N, and similarly for line (4.4). Also line (4.5) is replaced with

  (4.5) $\delta \leftarrow \delta_1 \delta_2$

As usual, this scheme can be shown to meet Definition 4.1. The security result is the following.

Theorem 4.9 *Let* $(\mathsf{K}_{\rm gap}, \mathsf{S}_{\rm ddh})$ *be a gap DH group specifier and let* $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ *be a SS scheme. Let* $\mathcal{Gap}$-$\mathcal{TS}$ *be the TS associated to them as defined above. If the one-more CDH problem associated to* $\mathsf{K}_{\rm gap}$ *is hard and* $\mathcal{SS}$ *is unforgeable under adaptive chosen-message attack, then* $\mathcal{Gap}$-$\mathcal{TS}$ *is transitively unforgeable under adaptive chosen-message attack.* ∎

The proof is very similar to that of Theorem 4.5. The one-more CDH adversary A gets input $\hat{\mathbb{G}}, g, q, u$ and runs the given forger F on input $tpk = (\hat{\mathbb{G}}, g, q, u, spk)$, where it generates (spk, ssk) via $\mathsf{SKg}(1^k)$. It then proceeds just like the one-more RSA adversary in the proof of Theorem 4.5, except that when the latter calls $\textsc{Inv}(\cdot)$, the one-more CDH adversary calls its $\textsc{Cdh}(\cdot)$ oracle, and operations modulo N are now operations in $\hat{\mathbb{G}}$.

4.7 Eliminating Node Certificates via Hashing

The above schemes use the node certification technique, and the standard signatures involved are a significant factor in the cost of the scheme. Here we show how, for some of the above schemes, one can eliminate node certificates by specifying the public label of a node i as the output of a hash function applied to i. No explicit certification is attached to this value. Rather, we will be able to show that the edge label provides an "implicit authentication" of the associated node label that suffices to be able to prove that the scheme is transitively unforgeable under adaptive chosen-message attack, in a model where the hash function is a random oracle. Let us now illustrate this by presenting the schemes.

4.7.1 The $\mathcal{RSAH}$-$\mathcal{TS}$ Scheme

To any RSA key generator $\mathsf{K}_{\rm rsa}$, we associate a transitive signature scheme $\mathcal{RSAH}$-$\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ defined as follows:

- $\mathsf{TKg}(1^k)$ does the following

  (1.1) Run $\mathsf{K}_{\rm rsa}(1^k)$ to get a triple (N, e, d)
  (1.2) Output $tpk = (N, e)$ as the public key and $tsk = (N, d)$ as the secret key.

  Now the following algorithms all have oracle access to a random function $\mathrm{H}_N\colon \mathbb{N} \to \mathbb{Z}_N^*$. Note that unlike the $\mathcal{RSA}$-$\mathcal{TS}$ scheme, the decryption exponent d is not discarded, but it is part of the secret key.

- The (stateless) signing algorithm TSign, when invoked on inputs tsk, i, j, meaning when asked to produce a signature on edge $\{i, j\}$, it does the following:

(2.1) If $i > j$ then swap(i, j)
(2.2) $\delta \leftarrow [\mathrm{H}_N(i)\mathrm{H}_N(j)^{-1}]^d \bmod N$
(2.3) Return δ as the signature of $\{i, j\}$.

- TVf, on input $tpk = (N, e)$, nodes i, j and a candidate signature δ, proceeds as follows:

(3.1) If $i > j$ then swap(i, j)
(3.2) If $\delta^e \equiv \mathrm{H}_N(i)\mathrm{H}_N(j)^{-1} \bmod N$ then return 1 else return 0.

- The composition algorithm Comp takes tpk, nodes i, j, k and signatures δ_1 and δ_2, and computes a composed signature for edge $\{i, k\}$ as follows:

(4.1) If $i > k$ then swap(i, k) ; swap(δ_1, δ_2)
(4.2) If $i > j$ then $\delta_1 \leftarrow \delta_1^{-1} \bmod N$
(4.3) If $j > k$ then $\delta_2 \leftarrow \delta_2^{-1} \bmod N$
(4.4) $\delta \leftarrow \delta_1\delta_2 \bmod N$
(4.5) Return δ as the signature for $\{i, k\}$.

As illustrated by Figure 4.1, this brings some significant performance gains over $\mathcal{RSA}$-$\mathcal{TS}$, particularly with regard to signature size. Regarding security, in the experiment $\mathbf{Exp}^{\text{tu-cma}}_{\mathcal{RSAH}\text{-}\mathcal{TS},\mathsf{F}}(k)$, we consider $\mathrm{H}_N\colon\ \mathbb{N} \to \mathbb{Z}_N^*$ to be chosen at random after the public and secret keys (defining N) have been chosen. The TSign, TVf, and Comp algorithms, as well as the adversary, then get oracle access to H_N. In this random oracle model, we have the following.

Theorem 4.10 *Let K_{rsa} be an RSA key generator and let $\mathcal{RSAH}$-$\mathcal{TS}$ be the transitive signature scheme associated to K_{rsa} as defined above. If the one-more RSA problem associated to K_{rsa} is hard, then $\mathcal{RSAH}$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attack in the random oracle model.* ∎

Proof: Suppose we are given a poly(k)-time adversary F for $\mathcal{RSAH}$-$\mathcal{TS}$. We construct a poly(k)-time one-more RSA adversary A such that for all k

$$\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{RSAH}\text{-}\mathcal{TS}\mathsf{F}}(k) \le \mathbf{Adv}^{\text{1m-rsa}}_{\mathsf{K}_{\text{rsa}},\mathsf{A}}(k)\ . \tag{4.12}$$

The theorem follows from the assumption that the one-more RSA-inversion problem associated to K_{rsa} is hard. It remains to describe A.

The one-more RSA adversary A gets inputs N, e and has access to an inversion oracle $\textsc{Inv}(\cdot)$ and a challenge oracle $\textsc{Chall}$. It lets $tpk = (N, e)$ and runs F on input tpk. It will itself provide answers, both to F's queries to its random oracle and to F's signature queries.

Adversary A uses the function $L(i)$ as a table that represents H_N. When a query $\mathrm{H}_N(i)$ is made by F, adversary A does the following:

If $i \notin V$ then $V \leftarrow V \cup \{i\}$; $L(i) \xleftarrow{R} \textsc{Chall}()$; $\Delta(i,i) \leftarrow 1$
Return $L(i)$ to F,

A answers F's signature queries as in the proof of Theorem 4.5, but omitting lines (5.4) and (5.7) and replacing line (5.18) by :

(5.18) $\quad \delta \leftarrow \Delta(i,j)$; Return δ to F.

At the end of its execution, F outputs a forgery δ' for edge $\{i',j'\}$. We can assume without loss of generality that F queried the hash oracle on i' and j' (and hence that $i',j' \in V$), because if it didn't A can query the hash oracle itself after F outputs its forgery. The remaining actions of A, and its analysis, are just as in the proof of Theorem 4.5, where the elements of V that were not involved in signature queries (but were queried to the random oracle only) are treated as singleton components. Also, the event **E** defined there does not exist here and one proceeds as in the case that **E** does not happen. This completes the proof. ∎

4.7.2 The $\mathcal{FactH}$-$\mathcal{TS}$ Scheme

To eliminate node certificates from the $\mathcal{Fact}$-$\mathcal{TS}$ scheme, it is natural to want to let $L(i) = \mathrm{H}_N(i)$, where H_N is some public hash function. When trying to implement this function in practice, however, the same problem emerges as when trying to transform factoring-based SI schemes into IBI schemes in Section 3.5.1. We could consider setting $L(i) = \mathrm{H}_N(i)^2 \bmod N$ where H_N has range $\mathbb{Z}_N^*$, but this reveals a square root of $L(i)$ which makes the scheme insecure. Instead, we let the signer choose N to be a Blum integer (i.e. $N = pq$ with p and q primes such that $p \equiv q \equiv 3 \bmod 4$). Now we will use as H_N a hash function with range $\mathbb{Z}_N^*[+1]$, and let $\ell(i)$ be a random square root of either $\mathrm{H}_N(i)$ or $-\mathrm{H}_N(i)$, whichever is a square. Since the Jacobi symbol can be computed in polynomial time given N, such a hash function can be easily built starting from a cryptographic hash function.

The $\mathcal{FactH}$-$\mathcal{TS}$ scheme. We associate to any given Blum modulus generator $\mathsf{K}_{\mathrm{blum}}$ (as defined in Section 2.3.1) a transitive signature scheme $\mathcal{FactH}\text{-}\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ defined as follows:

- TKg, on input 1^k, runs $\mathsf{K}_{\mathrm{blum}}(1^k)$ to obtain (N,p,q) and outputs $tpk = N$ as the public key and $tsk = (N,p,q)$ as the matching secret key. All the following algorithms are now assumed to have oracle access to a function $\mathrm{H}_N\colon\ \mathbb{N}^* \to \mathbb{Z}_N^*[+1]$.
- TSign maintains state $St = (V,\ell)$ where $V \subseteq \mathbb{N}$ is the set of all queried nodes and the function $\ell\colon\ V \to \mathbb{Z}_N^*$ assigns to each node $i \in V$ a secret

label $\ell(i) \in \mathbb{Z}_N^*$. When invoked on inputs tsk, i, j, meaning when asked to produce a signature on edge $\{i, j\}$, it does the following:

If $i > j$ then swap(i, j)
If $i \notin V$ then $V \leftarrow V \cup \{i\}$; $\ell(i) \stackrel{R}{\leftarrow} \sqrt{\pm \mathrm{H}_N(i)} \bmod N$
If $j \notin V$ then $V \leftarrow V \cup \{j\}$; $\ell(j) \stackrel{R}{\leftarrow} \sqrt{\pm \mathrm{H}_N(j)} \bmod N$
$\delta \leftarrow \ell(i)\ell(j)^{-1} \bmod N$,

where the notation $x \stackrel{R}{\leftarrow} \sqrt{\pm y} \bmod N$ means that x is chosen at random from the four square roots of y or $-y \bmod N$, whichever is a square modulo N. (These roots can be efficiently computed using the prime factors p and q.) Return δ as the signature on $\{i, j\}$.

- TVf, on input $tpk = N$, nodes i, j and a signature δ, first swaps i and j if $i > j$. It returns 1 if $\mathrm{H}_N(i) \cdot \mathrm{H}_N(j)^{-1} \equiv \pm\delta^2 \bmod N$ and returns 0 otherwise.
- The composition algorithm Comp is identical to that of $\mathcal{RSAH}$-$\mathcal{TS}$.

A proof by induction can be used to show the following.

Proposition 4.11 The $\mathcal{FactH}$-$\mathcal{TS}$ transitive signature scheme described above satisfies the correctness requirement of Definition 4.1. ∎

Computational costs. Since half of the elements in $\mathbb{Z}_N^*$ have Jacobi symbol $+1$, a hash function evaluation requires the computation of two Jacobi symbols on average, which takes time quadratic in $|N|$. Computing square roots, however, is cubic in $|N|$, so this will dominate the cost of generating signatures. Verification and composition of signatures involve multiplications, inverses and Jacobi symbols mod N, all of which are operations quadratic in $|N|$.

Security. We prove breaking the $\mathcal{FactH}$-$\mathcal{TS}$ scheme equivalent to factoring in the random oracle model. This means that in the experiment $\mathbf{Exp}^{\text{tu-cma}}_{\mathcal{FactH}\text{-}\mathcal{TS},\mathsf{F}}(k)$ used to define the advantage of an adversary F, the function H_N is assumed to be chosen at random from the space of all functions mapping $\{0,1\}^*$ to $\mathbb{Z}_N^*[+1]$. The result is stated as a theorem below.

Theorem 4.12 *Let K_{blum} be a Blum modulus generator. Let $\mathcal{FactH}$-$\mathcal{TS}$ be the associated transitive signature scheme as defined above. If the factoring problem associated to K_{blum} is hard, then $\mathcal{FactH}$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attack in the random oracle model.* ∎

Proof: Suppose we have a polynomial-time tu-cma forger F for $\mathcal{FactH}$-$\mathcal{TS}$. We will give a factoring algorithm A that uses F as a subroutine to factor composite numbers generated by K_{blum}. On input Blum integer N, A runs F on input $tpk = N$, answering its random oracle queries for node i as

If $i \notin V$ then
$\ell(i) \xleftarrow{R} \mathbb{Z}_N^*$; $s(i) \xleftarrow{R} \{-1,+1\}$; $V \leftarrow V \cup \{i\}$
Return $s(i) \cdot \ell(i)^2 \bmod N$ to F.

Half of the elements in $\mathbb{Z}_N^*[+1]$ are squares with Legendre symbols $+1$ modulo both p and q, while the other half are non-squares with Legendre symbols -1 modulo both p and q. For a Blum integer N, -1 belongs to the latter subset, and every non-square in $\mathbb{Z}_N^*[+1]$ can be written as the product of -1 times a square mod N. Consequently, the output of the above algorithm follows the same distribution as a truly random function from $\mathbb{N}$ to $\mathbb{Z}_N^*[+1]$, as required.
A answers F's signature queries as follows:

If $i \notin V$ then
$\ell(i) \xleftarrow{R} \mathbb{Z}_N^*$; $s(i) \xleftarrow{R} \{-1,+1\}$; $V \leftarrow V \cup \{i\}$
If $j \notin V$ then
$\ell(j) \xleftarrow{R} \mathbb{Z}_N^*$; $s(j) \xleftarrow{R} \{-1,+1\}$; $V \leftarrow V \cup \{j\}$
If $i < j$ then return $\ell(i) \cdot \ell(j)^{-1} \bmod N$
else return $\ell(j) \cdot \ell(i)^{-1} \bmod N$.

Let F's forgery be (i', j', δ'). Again, we assume without loss of generality that F queried the random oracle on i' and j' before halting. Let E be the set of edges for which F queried a signature and let $\widetilde{G} = (V, \widetilde{E})$ be the transitive closure of the graph $G = (V, E)$. If F's output is not a successful forgery, meaning that either $\mathsf{TVf}(N, i', j', \delta') \neq 1$ or $\{i', j'\} \in \widetilde{E}$, A aborts. If $i' < j'$ then let $\delta \leftarrow \ell(i') \cdot \ell(j')^{-1} \bmod N$, otherwise let $\delta \leftarrow \ell(j') \cdot \ell(i')^{-1} \bmod N$. If $\delta \equiv \pm\delta' \bmod N$, then A aborts, otherwise it outputs $\gcd(\delta + \delta', N)$.
By arguments analogous to those in the proof of Theorem 4.6, A is successful whenever it doesn't abort. The advantage of the forger F is bounded by

$$\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{FactH}\text{-}\mathcal{TS},\mathsf{F}}(k) \quad \leq \quad 2 \cdot \mathbf{Adv}^{\text{fact}}_{\mathsf{K}_{\text{blum}},\mathsf{A}}(k) \tag{4.13}$$

by a similar information-theoretic reasoning as in the proof of Theorem 4.6. ∎

4.7.3 The *GapH-TS* Scheme

The discrete logarithm-based *DL-TS* and *DL1m-TS* schemes are not amenable to a hash-based improvement because the discrete exponentiation function is not trapdoor. For the *Gap-TS* scheme on the other hand, one can view $a = \log_g(u)$ as trapdoor information allowing to compute secret labels from public labels, giving rise to the stateless and very compact (in terms of signature size) *GapH-TS* scheme described below.

- The key generation algorithm $\mathsf{TKg}(1^k)$ calls $\mathsf{K}_{\text{gap}}(1^k)$ to generate a cyclic group description $\hat{\mathbb{G}}$, its order q and a generator g. It chooses $a \xleftarrow{R} \mathbb{Z}_q$

and computes $u \leftarrow g^a$. It outputs the public key $tpk = (\hat{\mathbb{G}}, q, g, u)$ and the corresponding secret key $tsk = (\hat{\mathbb{G}}, q, g, a)$. All algorithms have oracle access to a random function $\mathrm{H}_{\hat{\mathbb{G}}} \colon \mathbb{N} \to \hat{\mathbb{G}}$.

- The (stateless) signing algorithm TSign, on input nodes i, j and secret key $tsk = (\hat{\mathbb{G}}, q, g, a)$, proceeds exactly as the TSign algorithm of $\mathcal{RSAH}$-$\mathcal{TS}$ but replacing line (2.2) by

  (2.2) $\quad \delta \leftarrow \left[\mathrm{H}_{\hat{\mathbb{G}}}(i) \cdot \mathrm{H}_{\hat{\mathbb{G}}}(j)^{-1}\right]^a.$

- TVf, on input $tpk = (\hat{\mathbb{G}}, q, g, u)$, nodes i, j and candidate signature δ, first swaps i and j if $i > j$. It outputs 1 if $\mathsf{S}_{\mathrm{ddh}}(\hat{\mathbb{G}}, q, g, u, \mathrm{H}_{\hat{\mathbb{G}}}(i)\mathrm{H}_{\hat{\mathbb{G}}}(j)^{-1}, \delta) = 1$, or returns 0 otherwise.

- The Comp algorithm is the same as that of $\mathcal{RSAH}$-$\mathcal{TS}$, except that the operations in lines (4.2), (4.3) and (4.4) are performed in $\hat{\mathbb{G}}$, rather than modulo N.

Note that just like the short signature scheme of Boneh et al. [BLS01], an edge signature under $\mathcal{GapH}$-$\mathcal{TS}$ contains only a single element of $\hat{\mathbb{G}}$, which can be represented in roughly 140 bits when using elliptic curves to achieve the same security as a 1024-bit RSA modulus [LV01].

Proposition 4.13 The $\mathcal{GapH}$-$\mathcal{TS}$ transitive signature scheme described above is correct according to Definition 4.1. ∎

Theorem 4.14 *Let* $(\mathsf{K}_{\mathrm{gap}}, \mathsf{S}_{\mathrm{ddh}})$ *be a Gap-DH group specifier and let* $\mathrm{H}_{\hat{\mathbb{G}}} : \mathbb{N} \to \hat{\mathbb{G}}$ *be a random oracle. The associated* $\mathcal{GapH}$-$\mathcal{TS}$ *transitive signature scheme described above is transitively unforgeable under adaptive chosen-message attack under the one-more Gap-DH assumption associated to* $\mathsf{K}_{\mathrm{gap}}$. ∎

The proof of the security statement is very similar to that of $\mathcal{RSAH}$-$\mathcal{TS}$: the one-more CDH adversary A gets $\hat{\mathbb{G}}, g, q, u$ as input and runs the forger F on input $tpk = (\hat{\mathbb{G}}, g, q, u)$. It then proceeds exactly as the one-more RSA adversary of Theorem 4.10, replacing all operations modulo N with operations in $\hat{\mathbb{G}}$, and replacing $\textsc{Inv}(\cdot)$ calls with calls to the $\textsc{Cdh}(\cdot)$ oracle.

4.7.4 A General Construction

The similarities between the different transitive signature schemes are so striking that one could imagine – or even expect – a more general framework to lie underneath. Indeed, following our work, Hohenberger [Hoh03] presented two general constructions for (undirected) TS schemes that are provably secure if

certain assumptions on the underlying algebraic group hold, but that make abstraction of the exact type of group being used. More specifically, she crystallizes the intuition of schemes having multiple valid edge labels for one edge explained on page 87 as *weakly collision-resistant non-injective group homomorphisms* and proves that breaking the associated node-certificate-using TS scheme is equivalent to either finding collisions in the group homomorphism, or breaking the underlying SS scheme. Both Micali and Rivest's $\mathcal{DL}$-$\mathcal{TS}$ scheme and our $\mathcal{Fact}$-$\mathcal{TS}$ schemes are captured under this definition. Furthermore, she constructs a second node-certificate-using TS scheme based on *one-way group isomorphisms* that is provably secure under an associated one-more inversion assumption and the security of the underlying SS scheme, thereby capturing the $\mathcal{RSA}$-$\mathcal{TS}$ and $\mathcal{DL1m}$-$\mathcal{TS}$ schemes. (Of these, only $\mathcal{RSA}$-$\mathcal{TS}$ was claimed by [Hoh03], since $\mathcal{DL1m}$-$\mathcal{TS}$ had not been proposed yet.) Strictly speaking, the $\mathcal{Gap}$-$\mathcal{TS}$ scheme is not covered by this definition because the homomorphism is not computable without knowing the secret key, but this can be fixed by weakening the computability requirement to a samplability requirement.

In this section, we present a general construction that encompasses all TS schemes obtained through our hash-based improvement ($\mathcal{RSAH}$-$\mathcal{TS}$, $\mathcal{FactH}$-$\mathcal{TS}$ and $\mathcal{GapH}$-$\mathcal{TS}$). One way of doing this would be to extend Hohenberger's definitions of weakly collision-resistant non-injective group homomorphisms and one-way group homomorphisms with a trapdoor notion, to construct a TS scheme from each of these definitions, and to prove the first secure under the collision-resistance of the homomorphism and the second under the associated one-more assumption. This approach however would involve writing two proofs from scratch, largely repeating the work of [Hoh03]. Instead, we put forward a single definition of *trapdoor samplable group homomorphisms*, construct two TS schemes based on it, one node-certification-based and the other hash-based, and show that the security of the latter follows from the security of the former. By observing that the $\mathcal{RSA}$-$\mathcal{TS}$, $\mathcal{Fact}$-$\mathcal{TS}$ and $\mathcal{Gap}$-$\mathcal{TS}$ schemes are secure under appropriate assumptions, the security of the $\mathcal{RSAH}$-$\mathcal{TS}$, $\mathcal{FactH}$-$\mathcal{TS}$ and $\mathcal{GapH}$-$\mathcal{TS}$ schemes under the same assumptions in the random oracle model follows.

In essence, trapdoor samplable group homomorphisms are a refinement of trapdoor samplable relations as defined in Definition 3.5, where the relation is a group homomorphism and an additional verifiability property is required. This is not surprising, since the random-oracle-using technique to transform node-certificate-based TS schemes into hash-based TS schemes is very similar to the technique used by the cSI-2-IBI transform of Construction 3.7 to turn SI schemes into IBI schemes. Both solve the problem of assigning a public value (the public label for TS schemes, part of a public key for IBI schemes) to an entity (a node in a graph for TS schemes, a user's identity for IBI schemes) by defining the value as the output of a hash function applied to the "name" of the entity, and compute the entity's secret using a piece of trapdoor information.

The restriction to group homomorphisms is due to the transitive property that requires algebraic manipulation of public labels; the verifiability property is used in the verification algorithms of cSI schemes, but was not explicitly needed in the cSI-2-IBI transform and hence it was not included in Definition 3.5.

Definition 4.15 A *family of trapdoor samplable group homomorphisms* $\mathcal{TH}$ is a quadruple of polynomial time algorithms $(\mathsf{THG}, \mathsf{THSample}, \mathsf{THVf}, \mathsf{THInv})$ where:

- THG is a randomized algorithm that on input 1^k outputs the description $\langle\psi\rangle$ of a group homomorphism $\psi : \mathbb{G}_0 \to \mathbb{G}_1$ together with a trapdoor t. We use multiplicative notation for both groups $\mathbb{G}_0$ and $\mathbb{G}_1$, and assume that multiplication and inversion of elements in both groups are efficiently computable given $\langle\psi\rangle$.
- the randomized $\mathsf{THSample}$ algorithm, on input $\langle\psi\rangle$, generates a tuple (x, y) such that x is uniformly distributed over $\mathbb{G}_0$ and $y = \psi(x)$.
- the verification algorithm THVf, on input $\langle\psi\rangle, x, y$, returns 1 if $y = \psi(x)$ and returns 0 otherwise.
- the THInv algorithm, on input a homomorphism description $\langle\psi\rangle$, the corresponding trapdoor t and an element $y \in \mathbb{G}_1$, returns a random element of $\psi^{-1}(y) = \{x \in \mathbb{G}_0 \mid \psi(x) = y\}$.
- all homomorphisms generated by THG are regular, meaning that $|\psi^{-1}(y)|$ is equal for all $y \in \mathbb{G}_1$.

∎

As in Definition 3.5, a requirement relating the hardness of inverting the homomorphism to the value of the security parameter is not needed, since this will be implied by the security of the originating node-certificate-based TS scheme. Also, we do not assume that the forward direction of the homomorphism be computable, either with or without the trapdoor information.

The following construction associates a node-certificate-based TS scheme to any family of trapdoor samplable group homomorphisms and any SS scheme.

Construction 4.16 Let $\mathcal{TH} = (\mathsf{THG}, \mathsf{THSample}, \mathsf{THVf}, \mathsf{THInv})$ be a family of trapdoor samplable group homomorphisms as per Definition 4.15, and let $\mathcal{SS} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ be a SS scheme. We associate to these a node-certificate-based TS scheme $\mathcal{NC}\text{-}\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ as follows:

- The key generation algorithm $\mathsf{TKg}(1^k)$ runs $\mathsf{THG}(1^k)$ to generate the description $\langle\psi\rangle$ of a homomorphism $\psi : \mathbb{G}_0 \to \mathbb{G}_1$ and corresponding trapdoor t, and runs $\mathsf{SKg}(1^k)$ to generate a standard signature key pair (spk, ssk). It outputs $tpk = (\langle\psi\rangle, spk)$ as the public key and $tsk = (\langle\psi\rangle, ssk)$ as the secret key. The trapdoor t is discarded.

- The TSign algorithm maintains as state information a set of nodes V and functions $\ell : V \to \mathbb{G}_0$, $L : V \to \mathbb{G}_1$ and $\Sigma : V \to \{0,1\}^*$. When invoked on inputs $tsk = (\langle\psi\rangle, ssk), i, j$, it proceeds just like the $\mathcal{RSA}$-$\mathcal{TS}$ algorithm, except for the following changes:

  (2.3) $V \leftarrow V \cup \{i\}$; $(\ell(i), L(i)) \xleftarrow{R} \mathsf{THSample}(\langle\psi\rangle)$
  (2.6) $V \leftarrow V \cup \{j\}$; $(\ell(j), L(j)) \xleftarrow{R} \mathsf{THSample}(\langle\psi\rangle)$
  (2.8) $\delta \leftarrow \ell(i)\ell(j)^{-1}$.

- The TVf algorithm, on input public key $tpk = (\langle\psi\rangle, spk)$, nodes i, j and candidate signature $\sigma = (C_1, C_2, \delta)$, is the same as the TVf algorithm of $\mathcal{RSA}$-$\mathcal{TS}$ with the last line is changed to

  (3.4) If $\mathsf{THVf}(\langle\psi\rangle, \delta, L_i L_j^{-1}) = 1$ then return 1 else return 0.

- The Comp algorithm is identical to that of $\mathcal{RSA}$-$\mathcal{TS}$, except that operations are performed in $\mathbb{G}_0$ instead of modulo N.

∎

We now argue that the $\mathcal{RSA}$-$\mathcal{TS}$, $\mathcal{F}act$-$\mathcal{TS}$ and $\mathcal{G}ap$-$\mathcal{TS}$ schemes are all special instances of $\mathcal{NC}$-$\mathcal{TS}$. This is true for the $\mathcal{RSA}$-$\mathcal{TS}$ scheme when considering homomorphism $\psi : \mathbb{Z}_N^* \to \mathbb{Z}_N^* : x \mapsto x^e \bmod N$ with description (N, e) and trapdoor d. Since the homomorphism is efficiently computable, sampling and verifying pairs is trivial. Inversion is done by raising to the power d modulo N. The regularity property is also trivial because ψ is an automorphism. The $\mathcal{F}act$-$\mathcal{TS}$ scheme can be seen as an instantiation of $\mathcal{NC}$-$\mathcal{TS}$ with $\psi : \mathbb{Z}_N^* \to QR_N : x \mapsto x^2 \bmod N$ with description N and trapdoor (p, q). Sampling and verifying pairs is done by computing the homomorphism, inversion is done by computing all four square roots and selecting one of them at random. Finally, the $\mathcal{G}ap$-$\mathcal{TS}$ scheme is identical to $\mathcal{NC}$-$\mathcal{TS}$ when instantiated with $\psi : \hat{\mathbb{G}} \to \hat{\mathbb{G}} : x \mapsto x^{(a^{-1} \bmod q)}$ with description $\langle\psi\rangle = (\hat{\mathbb{G}}, q, g, u)$ and trapdoor $a = \log_g u$. Sampling pairs can be done by choosing $r \xleftarrow{R} \mathbb{Z}_q$ and outputting (g^r, u^r), verifying if $y = \psi(x)$ for a given (x, y) is done by using the $\mathsf{S}_{\mathrm{ddh}}$ algorithm to check that (g, u, y, x) is a valid Diffie-Hellman tuple. The inverted homomorphism is computed by raising to the power a in $\hat{\mathbb{G}}$.

The following construction shows how to base a hash-based TS scheme on a family of trapdoor samplable group homomorphisms, capturing the $\mathcal{RSAH}$-$\mathcal{TS}$, $\mathcal{F}act\mathcal{H}$-$\mathcal{TS}$ and $\mathcal{G}ap\mathcal{H}$-$\mathcal{TS}$ as special cases.

Construction 4.17 We associate to any family of trapdoor samplable group homomorphisms $\mathcal{TH} = (\mathsf{THG}, \mathsf{THSample}, \mathsf{THVf}, \mathsf{THInv})$ a hash-based transitive signature scheme $\mathcal{H}$-$\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf})$ as follows:

- The key generation algorithm $\mathsf{TKg}(1^k)$ runs $\mathsf{THG}(1^k)$ to generate a pair $(\langle\psi\rangle, t)$. It outputs $tpk = \langle\psi\rangle$ as the public key $tsk = (\langle\psi\rangle, t)$ as the secret key. The scheme also assumes the availability of a random oracle $\mathrm{H} : \mathbb{N} \to \mathbb{G}_1$.
- The TSign algorithm maintains as state information a set of nodes V and a function $\ell : V \to \mathbb{G}_0$. To sign edge $\{i, j\}$, it proceeds as follows:

  If $i > j$ then swap(i, j)
  If $i \notin V$ then $\ell(i) \xleftarrow{R} \mathsf{THInv}(\langle\psi\rangle, t, \mathrm{H}(i))$
  If $j \notin V$ then $\ell(j) \xleftarrow{R} \mathsf{THInv}(\langle\psi\rangle, t, \mathrm{H}(j))$
  $\delta \leftarrow \ell(i)\ell(j)^{-1}$; Return δ.

- On input public key $tpk = \langle\psi\rangle$, nodes i, j and candidate signature δ as input, the verification algorithm TVf returns 1 if $\mathsf{THVf}(\langle\psi\rangle, \delta, \mathrm{H}(i)\mathrm{H}(j)^{-1}) = 1$ and returns 0 otherwise.
- The Comp algorithm, on input public key $tpk = \langle\psi\rangle$, signatures δ_1, δ_2 and nodes i, j, k, computes a composed signature for $\{i, k\}$ as:

  If $i > k$ then swap(i, k) ; swap(δ_1, δ_2)
  If $i > j$ then $\delta_1 \leftarrow \delta_1^{-1}$
  If $j > k$ then $\delta_2 \leftarrow \delta_2^{-1}$
  $\delta \leftarrow \delta_1 \cdot \delta_2$; Return δ.

∎

The *RSAH-TS* and *GapH-TS* schemes are instances of *H-TS* for the same reasons that *RSA-TS* and *Gap-TS* are instances of *NC-TS*. The case of *Fact-TS* asks for a bit more explanation due to the small changes we had to apply to the scheme to make the random oracle implementable in the real world. Consider the group $\mathbb{G}$ of two-element sets $\{a, -a \bmod N\}$ where $a \in QR_N$ for a Blum modulus N, and with group operation $\{a, -a \bmod N\} \cdot \{b, -b \bmod N\} = \{ab \bmod N, -ab \bmod N\}$. The *FactH-TS* scheme can be seen as a "compact" but equivalent variant of the *H-TS* scheme when instantiated with homomorphism $\psi : \mathbb{Z}_N^* \to \mathbb{G} : x \mapsto \{x^2 \bmod N, -x^2 \bmod N\}$. Indeed, the public labels in the *FactH-TS* scheme are only a single element of $\mathbb{Z}_N^*$, but can be seen as a compact representation of the set containing both the element itself and its negative. The corresponding node-certificate-based scheme (the instantiation of *NC-TS* with the above homomorphism ψ) is easily seen to be secure, because signatures under it contain even less information than in the *Fact-TS* scheme, while the relaxed verification equation at most doubles the advantage of an adversary. As a consequence, the following theorem applies to all three hash-based TS schemes.

Theorem 4.18 *Let $\mathcal{NC}$-$\mathcal{TS}$ be the certificate-based transitive signature scheme associated to a familty of trapdoor samplable group homomorphisms $\mathcal{TH}$ as per Construction 4.16, and let $\mathcal{H}$-$\mathcal{TS}$ be the hash-based TS scheme associated to $\mathcal{TH}$ as per Construction 4.17. If $\mathcal{NC}$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attack, then $\mathcal{H}$-$\mathcal{TS}$ is transitively unforgeable under adaptive chosen-message attack in the random oracle model.* ∎

Proof: Given adversary A attacking $\mathcal{H}$-$\mathcal{TS}$, we construct an adversary F attacking $\mathcal{NC}$-$\mathcal{TS}$ as follows. Algorithm F is given input $tpk = (\langle\psi\rangle, spk)$ and has access to a signing oracle $\text{TSIGN}_{\mathcal{NC}\text{-}\mathcal{TS}}(\cdot,\cdot)$. (To avoid confusion between A's and F's signing oracles, we write the scheme in subscript.) F's strategy will be to simulate A's environment by using public labels in signatures from the $\text{TSIGN}_{\mathcal{NC}\text{-}\mathcal{TS}}$ oracle as answers to A's hash queries. A small technical problem that must be overcome, however, is that A can query for the hash value of a node i before it was involved in a signature query, while the only way for F to obtain the public label of a node i is by querying for the signature on a "dummy" edge $\{i, j\}$. To make sure that A's forgery does not coincide with a dummy edge, each node in A's view corresponds to a pair of nodes in F's view. Algorithm F maintains a "renaming" function $R : \mathbb{N} \to \mathbb{N}$ that represents the mapping from A's nodes to F's nodes. More specifically, F initializes set V to $\emptyset$ and runs A on input $\langle\psi\rangle$, answering its random oracle queries $\text{H}(i)$ as:

(1.1) If $i \notin V$ then
(1.2) $\quad V \leftarrow V \cup \{i\}$; $R(i) \leftarrow 2 \cdot |V|$ // *node renaming function*
(1.3) $\quad (C_1, C_2, \delta) \leftarrow \text{TSIGN}_{\mathcal{NC}\text{-}\mathcal{TS}}(R(i), R(i)+1)$
(1.4) $\quad$ Parse C_1 as $(R(i), L_i, \Sigma_i)$; $L(i) \leftarrow L_i$
(1.5) Return $L(i)$

and answering A's $\text{TSIGN}_{\mathcal{H}\text{-}\mathcal{TS}}(i, j)$ queries as:

(2.1) If $i \notin V$ then
(2.2) $\quad V \leftarrow V \cup \{i\}$; $R(i) \leftarrow 2 \cdot |V|$
(2.3) $\quad (C_1, C_2, \delta) \leftarrow \text{TSIGN}_{\mathcal{NC}\text{-}\mathcal{TS}}(R(i), R(i)+1)$
(2.4) $\quad$ Parse C_1 as $(R(i), L_i, \Sigma_i)$; $L(i) \leftarrow L_i$
(2.5) If $j \notin V$ then
(2.6) $\quad V \leftarrow V \cup \{j\}$; $R(j) \leftarrow 2 \cdot |V|$
(2.7) $\quad (C_1, C_2, \delta) \leftarrow \text{TSIGN}_{\mathcal{NC}\text{-}\mathcal{TS}}(R(j), R(j)+1)$
(2.8) $\quad$ Parse C_1 as $(R(j), L_j, \Sigma_j)$; $L(j) \leftarrow L_j$
(2.9) $(C_1, C_2, \delta) \leftarrow \text{TSIGN}_{\mathcal{NC}\text{-}\mathcal{TS}}(R(i), R(j))$
(2.10) If $((i > j)$ and $(R(i) > R(j)))$ or $((i < j)$ and $(R(i) < R(j)))$
(2.11) then return δ else return δ^{-1}

When A outputs its forgery δ for edge $\{i, j\}$ (assuming without loss of generality that $i < j$), then F outputs δ as its own forgery for edge $\{R(i), R(j)\}$ if $R(i) <$

$R(j)$, or δ^{-1} if $R(i) > R(j)$. If A did not previously query its $\text{TSIGN}_{\mathcal{H}\text{-}\mathcal{TS}}$ oracle for a signature of $\{i, j\}$, then F definitely never queried its $\text{TSIGN}_{\mathcal{NC}\text{-}\mathcal{TS}}$ oracle for a signature of $\{R(i), R(j)\}$ on line (2.9), and neither did it do so anywhere else, because the queries on lines (1.3), (2.3) and (2.7) always involve one even and one odd node label, while $R(i)$ and $R(j)$ are both even. In conclusion, F's forgery is valid if A's forgery is valid, and hence we can bound the advantage of A as

$$\mathbf{Adv}^{\text{tu-cma}}_{\mathcal{H}\text{-}\mathcal{TS},\mathsf{A}}(k) \leq \mathbf{Adv}^{\text{tu-cma}}_{\mathcal{NC}\text{-}\mathcal{TS},\mathsf{F}}(k).$$

∎

Combining Theorem 4.18 with the security results of the *RSA-TS*, *Fact-TS* and *Gap-TS* schemes as stated in Theorems 4.5, 4.6 and 4.9, yields that the *RSAH-TS*, *FactH-TS* and *GapH-TS* schemes are transitively unforgeable under adaptive chosen-message attack in the random oracle model assuming the hardness of the one-more RSA, factoring and one-more CDH problems, respectively.

4.8 Conclusion

In this chapter, we revisited the concept of transitive signatures as introduced by Micali and Rivest [MR02b]. We started off by answering an open question raised by Micali and Rivest regarding the security of an RSA-based scheme against *adaptive* adversaries. We then introduced new provably secure schemes based on factoring and Gap Diffie-Hellman groups, and proposed a hash-based technique that eliminates the need for so-called node certificates, thereby drastically reducing the signature length. We applied this technique to the schemes based on RSA, factoring and Gap-DH groups, and we further generalized the technique to any family of trapdoor samplable group homomorphism.

Part of the results contained in this chapter were published in the proceedings of the ASIACRYPT 2002 conference [3].

Chapter 5

Conclusion

In this thesis, we presented new provably secure schemes for a number of cryptographic problems, and provided security proofs for existing schemes that lacked such proof. We employed commonly accepted security notions wherever possible, and in the few occasions that we found such notions to be missing, we defined new, meaningful yet feasible security notions. We also presented abstract constructions and transformations that greatly simplify the task of proving the security of concrete schemes, and that moreover contribute to our understanding of the general principles underlying related schemes.

More specifically, we discussed identity-based identification and signature schemes in Chapter 3, and transitive signature schemes in Chapter 4. We summarize our contributions to these areas in Sections 5.1 and 5.2, respectively. We discuss a couple of interesting open problems in Section 5.3.

5.1 Identity-based Identification Schemes

A standard identification (SI) scheme enables a prover knowing secret key sk to interactively identify itself to a verifier knowing the corresponding public key pk, while eavesdroppers or even active adversaries cannot deduce any useful information from the conversation. To implement public-key cryptography in practice, one has to provide a secure way to associate individual users to their public keys, a problem that is commonly solved by setting up an expensive public-key infrastructure. Identity-based cryptography [Sha84] alleviates this problem by using the user's identity as his public key, and having the corresponding secret key usk delivered to the user by a central trusted key generation center.

Security notions. While numerous identity-based identification (IBI) schemes have been proposed in the literature, and while solid security notions do exist for SI schemes, prior to our work no security notions were known for IBI schemes

that take into account attacks from inside users of the system, possibly even colluding with other users. Consequently, all known IBI schemes were proven secure in weaker models, if at all.

We filled this somewhat surprising hiatus by providing adequate security notions for IBI schemes. We first paused to show this definition is easily met by a certificate-based IBI scheme based on an SI scheme and a standard signature (SS) scheme. This points to the fact that, unlike identity-based encryption schemes [BF01], trival solutions exist for IBI schemes, and that the goal of special-purpose IBI schemes is to outperform the trivial solution.

General transforms. Instead of manually rewriting a proof from scratch for every IBI scheme ever proposed, we defined a class of SI schemes called convertible SI (cSI) schemes and demonstrated a general random-oracle-using transformation cSI-2-IBI that when applied to a secure cSI scheme results in a secure IBI scheme (see Theorem 3.8).

Through known transformations by Fiat and Shamir [FS86] and Dodis et al. [DKXY03], we build an entire framework encompassing SI, IBI, SS and IBS schemes. We view schemes as occurring in families, as depicted in Figure 3.1. Under certain conditions, a cSI scheme *Name-SI* can be transformed into a SS scheme *Name-SS* through the fs-I-2-S transform [FS86], which in turn can be transformed into an IBS scheme *Name-IBS* through a generalization of the transform of Dodis et al. [DKXY03] that we call the cSS-2-IBS transform, all while preserving security. The *Name-IBS* scheme thus obtained coincides with the scheme obtained by applying the fs-I-2-S transform to the IBI scheme *Name-IBI* = cSI-2-IBI(*Name-SI*) resulting from our transform, thereby completing the picture of Figure 3.1.

While the above relations imply that the fs-I-2-S transformation is security-preserving when applied to an IBI scheme that was obtained as the cSI-2-IBI transform of a cSI scheme, we observed that this is not true for IBI schemes in general. We fixed this problem by presenting the efs-IBI-2-IBS transform and proving that it does preserve security for general IBI schemes.

Applying the framework. These tools in hand, we went through two decades of literature on SI, IBI, SS and IBS schemes, delivering security proofs for previously unproven schemes and for new schemes that we surface from these. An overview of our results is given in Table 3.2.

We found that a large number of schemes based on the hardness of factoring [FS86, FFS88, OO90, OS90] are actually special cases of a more general family of schemes that we called the *ItR* (for "iterated root") family. Their security as IBI schemes follows from our transform. We also found another scheme based on factoring, the *FF-SI* scheme, to be amenable to the identity-based setting through our transform.

From the famous RSA-based IBS scheme by Shamir [Sha84], we surfaced the *Sh-SI* scheme and proved that it is secure under passive attack, but in-

secure under active attack. This is sufficient, however, to prove the security of the $\mathcal{Sh}$-$\mathcal{IBS}$ scheme through our framework, which has been an open problem since its introduction in 1984. We also present a modified $\mathcal{Sh}^*$-$\mathcal{SI}$ scheme that is secure under active and concurrent attacks as well. The security of the $\mathcal{GQ}$-$\mathcal{SI}$ [GQ89] and $\mathcal{OKRSA}$-$\mathcal{SI}$ [Oka93] schemes was already well-studied, and extends to the identity-based case through our framework. We found one last RSA-based scheme by Girault [Gir90, SSN98] to be insecure, and showed an attack breaking all schemes in the family.

Following the renewed interest in identity-based cryptography after the introduction of pairings on elliptic curves to cryptography [JN03], new pairing-based IBS schemes have been proposed [SOK00, CC03, Yi03, Pat02, Hes03]. Barring the scheme of Cha and Cheon [CC03], none of these were proven secure in a full-fledged identity-based model (although the security of $\mathcal{Hs}$-$\mathcal{IBS}$ [Hes03] follows from Dodis et al. [DKXY03]). We surfaced the provably secure $\mathcal{Hs}$-$\mathcal{SI}$ and $\mathcal{ChCh}$-$\mathcal{SI}$ schemes such that their IBS siblings coincide with the schemes presented by Hess [Hes03] and independently by Cha and Cheon [CC03] and Yi [Yi03], respectively. We also surfaced the $\mathcal{SOK}$-$\mathcal{SI}$ scheme of which the IBS sibling is a close relative of, but not identical to the scheme presented by Sakai et al. [SOK00]. We proved the $\mathcal{SOK}$-$\mathcal{SI}$ scheme to be secure under passive attack, from which the security of $\mathcal{SOK}$-$\mathcal{IBS}$ follows, but the security of the original scheme [SOK00] remains open.

Due to their lack of trapdoors, discrete logarithm groups are not an obvious choice to construct IBI schemes. Nevertheless, Beth [Bet88] did propose such an IBI scheme, without providing any security proofs however. We surfaced the $\mathcal{Beth}^t$-$\mathcal{SI}$ scheme of which the IBI sibling coincides with the scheme of Beth [Bet88], but we were only able to prove the special case of $\mathcal{Beth}^1$-$\mathcal{SI}$ secure under passive attack under an unusual assumption on ElGamal signatures. From this, the passive security of $\mathcal{Beth}^1$-$\mathcal{IBI}$ and the security of $\mathcal{Beth}^1$-$\mathcal{SS}$ and $\mathcal{Beth}^1$-$\mathcal{IBS}$ under the same assumption follow.

Schemes needing direct proof. The only IBI scheme we found in the literature not originating from a cSI scheme is a scheme by Okamoto [Oka93] that we call $\mathcal{OKDL}$-$\mathcal{IBI}$. While no security proof for it was known prior to our work, we were able to prove it secure under the discrete logarithm assumption. We obtained a corresponding IBS scheme through our extended efs-IBI-2-IBS transform. Lastly, we presented a more natural variant called the $\mathcal{XDL}$-$\mathcal{IBI}$ that we also proved secure as an IBI scheme directly.

5.2 Transitive Signatures

In Chapter 4, we discussed transitive signature (TS) schemes as introduced by Micali and Rivest [MR02b], allowing to sign edges in a graph such that from two

signatures σ_1 and σ_2 on adjacent edges $\{i, j\}$ and $\{j, k\}$, anyone can compute a third signature σ_3 on the direct edge $\{i, k\}$.

THE SCHEMES. Apart from introducing the concept, Micali and Rivest also proposed the first non-trivial construction based on discrete logarithms, that we refer to as the $\mathcal{DL}$-$\mathcal{TS}$ scheme. Our starting point was the RSA-based $\mathcal{RSA}$-$\mathcal{TS}$ scheme that was briefly mentioned by Micali and Rivest to be secure against non-adaptive adversaries. We revisited this scheme and provided a security proof against adaptive adversaries, but under the one-more RSA assumption instead of its mere one-wayness.

In quest of a TS scheme that is provably secure against adaptive adversaries under the one-wayness of RSA, we presented the $\mathcal{Fact}$-$\mathcal{TS}$ scheme and proved it secure under the even weaker factoring assumption. The security proof involved a delicate information-theoretic lemma showing that signatures do not leak any relevant information about the secret choices of the signer. We also presented the $\mathcal{DL}1m$-$\mathcal{TS}$ scheme, a more natural and slightly more efficient variant of $\mathcal{DL}$-$\mathcal{TS}$ based on the one-more discrete logarithm problem, and the $\mathcal{Gap}$-$\mathcal{TS}$ scheme based on the one-more Gap Diffie-Hellman assumption.

ELIMINATING NODE CERTIFICATES. All of the above schemes follow a common paradigm that we called the *node certification paradigm*, in which a public label is assigned to each node by signing the node name together with its public label using a standard signature scheme. We presented a hash-based technique that eliminates the need for node certificates, and completely removes the standard signature scheme and its associated costs from the picture. The technique was easily applied to the $\mathcal{RSA}$-$\mathcal{TS}$ and $\mathcal{Gap}$-$\mathcal{TS}$ schemes to yield the $\mathcal{RSAH}$-$\mathcal{TS}$ and $\mathcal{GapH}$-$\mathcal{TS}$ schemes, and after minor modifications the $\mathcal{Fact}$-$\mathcal{TS}$ scheme was amenable to a hash-based variant $\mathcal{FactH}$-$\mathcal{TS}$ as well. All hash-based schemes were proven secure in the random oracle model.

Viewing the similarities between the different hash-based schemes, and continuing the line of generalized TS constructions of Hohenberger [Hoh03], we distilled the concept of *trapdoor samplable group homomorphisms* – which is remarkably close to the concept of trapdoor samplable relations on which we based the definition of cSI schemes in Chapter 3. From this concept, we constructed one TS scheme $\mathcal{NC}$-$\mathcal{TS}$ following the node-certification paradigm, and a second hash-based TS scheme called $\mathcal{H}$-$\mathcal{TS}$. We then showed that if $\mathcal{NC}$-$\mathcal{TS}$ is transitively unforgeable under chosen-message attack, then so is $\mathcal{H}$-$\mathcal{TS}$ in the random oracle model. Through this general theorem, the security of the $\mathcal{RSAH}$-$\mathcal{TS}$, $\mathcal{FactH}$-$\mathcal{TS}$ and $\mathcal{GapH}$-$\mathcal{TS}$ schemes is implied by the security of the $\mathcal{RSA}$-$\mathcal{TS}$, $\mathcal{Fact}$-$\mathcal{TS}$ and $\mathcal{Gap}$-$\mathcal{TS}$ schemes, respectively.

DEFINITIONAL CONTRIBUTIONS. We demonstrated a problem with the correctness definition of TS schemes as put forward by Micali and Rivest by showing that neither the $\mathcal{DL}$-$\mathcal{TS}$ schemes nor any of our schemes meets it. We presented a

new, more realistic correctness definition that is met by practical schemes, while avoiding the entanglement of security and correctness as in the old definition.

5.3 Open Problems

We conclude this thesis with a number of suggestions for further research.

Filling the gaps in Table 3.2. In spite of the considerable effort spent on investigating the security properties of schemes under different notions in Section 3.5, a few entries in Table 3.2 remain unanswered. The first concerns the security of the $\mathit{It\mathcal{R}\text{-}SI}$ scheme under concurrent attack. One would have to go through the details of the proof under active attack [Sch96] to verify whether the same proof strategy extends to concurrent attacks. The security of $\mathit{It\mathcal{R}\text{-}IBI}$ would follow from that of $\mathit{It\mathcal{R}\text{-}SI}$ by applying our framework.

A second series of open entries in the table is due to the $\mathcal{B}\mathit{eth}^t$ family. We proved the imp-pa security of the $\mathcal{B}\mathit{eth}^1\text{-}\mathcal{SI}$ scheme under a weak assumption on hashed-message ElGamal signatures. We were unable to extend the proof technique of Theorem 3.21 to active and concurrent attacks, as it is unclear how to simulate interactive prover sessions with the cheating verifier without knowing the secret key. Also, it is still an open problem if the $\mathcal{B}\mathit{eth}^1\text{-}\mathcal{SI}$ scheme can be proven secure under a more natural assumption.

The situation for the $\mathcal{B}\mathit{eth}^t$ family for $t > 1$ is even more peculiar, as we don't even know whether the $\mathcal{B}\mathit{eth}^t\text{-}\mathcal{SI}$ scheme is convertible. The sampling algorithm for $t = 1$ exploits the existential forgeability of textbook-ElGamal signatures, but this approach fails for higher key multiplicities because a single value for R has to "fit" all t values of X_i. The same problem occurs when trying to simulate conversations for the $\mathcal{B}\mathit{eth}^t\text{-}\mathcal{SI}$ scheme, leaving even security under passive attack as an open problem.

Tighter reductions through direct proofs. The quadrangle of transformations in Figure 3.1 is a powerful tool that greatly simplifies the task of proving the asymptotical security of IBI and IBS schemes, but its generality prevents scheme-specific optimizations that would result in tighter security reductions. The need for such optimizations becomes clear when filling in concrete values in the reduction equations of the general transforms. For example, to make the $\mathcal{S}\mathit{h}\text{-}\mathcal{IBS}$ scheme as secure as the $\mathcal{S}\mathit{h}\text{-}\mathcal{SI}$ scheme with a 1024-bit modulus against an adversary who is allowed 2^{60} queries to both of its random oracles and 2^{30} signature queries, according to Equations (3.3) and (3.5) one would have to instantiate it with a 6701-bit modulus. If we want to make it as secure as the 1024-bit RSA problem itself and we hence also have to take into account the square root of Equation (3.7) due to the Reset Lemma, then the equivalent modulus length even runs up to 19611 bits. However, this does *not* mean that the scheme is insecure for smaller moduli: the theorems don't imply an actual

attack, and tighter proofs might exist.

In particular, for some IBI and IBS schemes it might be possible to eliminate the factor $\mathbf{Q}_{\overline{\mathsf{CV}}}^{\mathrm{H}}$ induced by Equation (3.4) when proving security directly, instead of through the transform. For example, when proving security of IBI schemes under a one-more assumption (e.g. the $\mathcal{GQ}$-$\mathcal{IBI}$, $\mathcal{Sh}$-$\mathcal{IBI}$, $\mathcal{Sh}^*$-$\mathcal{IBI}$, $\mathcal{SOK}$-$\mathcal{IBI}$ and $\mathcal{Hs}$-$\mathcal{IBI}$ schemes under active and concurrent attack), one could use the challenge oracle instead of the Sample algorithm to generate answers to $\overline{\mathsf{A}}$'s random oracle queries. This would allow to transform an attack on *any* identity into a solution of the problem (instead of only the identity that was guessed in advance), thereby eliminating the factor $\mathbf{Q}_{\overline{\mathsf{CV}}}^{\mathrm{H}}$ from the reduction equation.

Identity-based cryptography without random oracles. The fact that both the $\mathsf{cSI\text{-}2\text{-}IBI}$ and the $\mathsf{cSS\text{-}2\text{-}IBS}$ transform have security proofs in the random oracle model does not appear to be a coincidence. Trivial solutions like the certificate-based IBI scheme of Section 3.3 left apart, all currently known identity-based schemes need a random oracle to map the (non-random) identity string to a random element of some set. While reasonably efficient schemes not needing random oracles have been proposed for other primitives like standard encryption [CS98] and signatures [CS00, GHR99], such schemes only exist under the form of trivial solutions (e.g. IBI and IBS schemes), or remain elusive altogether (e.g. identity-based encryption).

Viewing the objections against the random oracle model that we formulated in Section 2.2, it would be interesting to investigate the existence of practical and efficient identity-based cryptography in the standard model. Such solutions might involve alternative assumptions on hash functions (such as the *division intractability* [GHR99]) that go beyond mere collision-resistance, but that are still more reasonable than the random oracle model.

Directed transitive signatures. All transitive signature schemes we suggested in Chapter 4 are for undirected graphs only. If truly compelling applications of transitive signatures exist, however, they are more likely to be found for directed graphs than for the undirected case. At this point, no constructions for directed transitive signatures have been proposed, and Hohenberger [Hoh03] even provided evidence that they might be very hard to construct.

The problem can already be seen from trying to convert currently known schemes to the undirected setting: on the one hand, we still need an algebraic operation, say multiplication, on node labels to provide composition of signatures, but on the other hand it should be hard to invert node labels. Hohenberger takes this idea a couple of steps further and proves that the existence of a directed transitive signature scheme would imply the existence of a special algebraic structure, called an *Abelian trapdoor group with infeasible inversion*, that is not known to exist.

Hohenberger's result, however, applies only to schemes that follow the node certification paradigm, as her model sees node certification as an intrinsic func-

tionality of a transitive signature scheme. It is not unthinkable that directed transitive signature schemes exist using a completely different approach, without needing to imply the existence of an exotic algebraic structure.

Compressing certificate chains. One might try to apply transitive signatures to shrink down so-called *certificate chains* to a single signature. Certificate chains are used to trace back the authenticity of a user's public key to a *root certificate* that is typically embedded in the verifier's software. They arise from hierarchically structured PKIs in which each certification authority (CA) signs the public key of the next. A certificate chain of length n tracing back a user's public key pk_n to a root public key pk_0 contains n signatures and n public keys, as follows:

$$pk_n \parallel \mathsf{Sign}(sk_{n-1}, pk_n) \parallel pk_{n-1} \parallel \mathsf{Sign}(sk_{n-2}, pk_{n-1}) \parallel \ldots \parallel pk_1 \parallel \mathsf{Sign}(sk_0, pk_1).$$

In spite of the first-sight analogy between graphs and CA trees, transitive signatures cannot help to compress this chain into a single signature, as the signatures that need to be composed here are signed under different secret keys, while transitive signatures are limited to a single signer.

So-called *aggregate signatures* [BGLS03, LMRS] are better suited for the job, as they combine n signatures of n signers on n different messages into a single signature of constant length. This indeed allows to compress the n signatures above into a single signature, but unfortunately *all* public keys in the chain are needed to verify the signature, resulting in a significantly reduced but still linear-length certificate chain

$$pk_n \parallel pk_{n-1} \parallel \ \ldots \ \parallel pk_1 \parallel \sigma \ .$$

Ultimately, we would like to reduce the chain even further to something of the form

$$pk_n \parallel \sigma,$$

where σ can be verified using pk_n and pk_0 only. We need a primitive with a special kind of composition that allows to "squeeze a key pair from the middle", meaning that given a signature for message M under sk_1 and a signature for pk_1 under sk_2, it should be possible to compute a third signature for message M under sk_2 directly. No construction offering such functionality is currently known.

Bibliography

[AABN02] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In L. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 418–433. Springer-Verlag, April 2002.

[Adl79] Leonard M. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography (abstract). In IEEE, editor, *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, pages 55–60. IEEE Computer Society Press, 1979.

[AR00] Michel Abdalla and Leonid Reyzin. A new forward-secure digital signature scheme. In T. Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 116–129. Springer-Verlag, 2000.

[BBB+00] Olivier Baudron, Fabrice Boudot, Philippe Bourel, Emmanuel Bresson, Johann Corbel, Laurent Frisch, Henri Gilbert, Marc Girault, Louis Goubin, Jean-François Misarsky, Phong Nguyen, Jacques Patarin, David Pointcheval, Guillaume Poupard, Jacques Stern, and Jacques Traoré. GPS – an asymmetric identification scheme for on the fly authentication of low cost smart cards. NESSIE Submission, November 2000. Available from `http://www.cryptonessie.org`.

[BD89] Mike Burmester and Yvo Desmedt. Remarks on soundness of proofs. *Electronics Letters*, 25(22):1509–1511, 1989.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology – CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer-Verlag, 1998.

[Bel98] Mihir Bellare. Practice-oriented provable security. In Eiji Okamoto, George I. Davida, and Masahiro Mambo, editors, *Proceedings of First International Workshop on Information Security (ISW 97)*, volume 1396 of *Lecture Notes in Computer Science*, pages 221–231. Springer-Verlag, 1998.

[Ben87] Josh Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, 1987.

[Bet88] Thomas Beth. Efficient zero-knowledged identification scheme for smart cards. In C. Gunther, editor, *Advances in Cryptology – EUROCRYPT 1988*, volume 330 of *Lecture Notes in Computer Science*, pages 77–86. Springer-Verlag, May 1988.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag, August 2001.

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432. Springer-Verlag, 2003.

[Ble96] Daniel Bleichenbacher. Generating ElGamal signatures without knowing the secret key. In U. Maurer, editor, *Advances in Cryptology – EUROCRYPT 1996*, Lecture Notes in Computer Science, pages 10–18. Springer-Verlag, 1996.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer-Verlag, 2001.

[BM92] Ernest F. Brickell and Kevin S. McCurley. An interactive identification scheme based on discrete logarithms and factoring. *Journal of Cryptology*, 5(1):29–39, 1992.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer-Verlag, December 2000.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, 2003.

[Bol03a] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In Y. Desmedt, editor, *Advances in Cryptology – Public-Key Cryptography 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer-Verlag, January 2003.

[Bol03b] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme. In Y. Desmedt, editor, *Advances in Cryptology – Public-Key Cryptography 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer-Verlag, 2003.

[Bon99] Dan Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the American Mathematical Society (AMS)*, 46(2):203–213, 1999.

[BP02] Mihir Bellare and Adriana Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attack. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 162–177. Springer-Verlag, August 2002.

[BR93a] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In ACM, editor, *Proceedings of the 1st Conference on Computer and Communications Security*, pages 62–73. ACM Press, November 1993.

[BR93b] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures - How to sign with RSA and Rabin. In U. Maurer, editor, *Advances in Cryptology – EUROCRYPT 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer-Verlag, 1996.

[BR98] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In A. De Santis, editor, *Advances in Cryptology – EURO-*

CRYPT 1994, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer-Verlag, 1998.

[CC03] Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap diffie-hellman groups. In Y. Desmedt, editor, *Advances in Cryptology – Public-Key Cryptography 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30. Springer-Verlag, January 2003.

[CEvdG88] David Chaum, Jan-Hendrik Evertse, and Jeroen van de Graaff. An improved protocol for demonstrating possession of discrete logarithms and some generalizations. In D. Chaum, editor, *Advances in Cryptology – EUROCRYPT 1987*, volume 304 of *Lecture Notes in Computer Science*, pages 127–141. Springer-Verlag, 1988.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle model, revisited. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 209–218. ACM Press, 1998.

[Cha83] David Chaum. Blind signatures for untraceable payments. In D. Chaum, R. Rivest, and A. Sherman, editors, *Advances in Cryptology: Proceedings of CRYPTO '82*, pages 199–203. Plenum Press, 1983.

[CK01] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In B. Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474. Springer-Verlag, 2001.

[CKN03] Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing chosen-ciphertext security. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 565–582. Springer-Verlag, 2003.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *Cryptography and Coding, 8th IMA International Conference, Cirencester, UK, December 17-19, 2001, Proceedings*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer-Verlag, 2001.

[Cor00] Jean-Sébastien Coron. On the exact security of full domain hash. In M. Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, Lecture Notes in Computer Science, pages 229–235. Springer-Verlag, 2000.

[CRR02] Suresh Chari, Tal Rabin, and Ronald Rivest. An efficient signature scheme for route aggregation. Manuscript, available from `http://theory.lcs.mit.edu/~rivest/publications.html`, 2002.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *Advances in Cryptology – CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer-Verlag, 1998.

[CS00] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, 2000.

[DBP96] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. RIPEMD-160: A strengthened version of RIPEMD. In D. Gollmann, editor, *Fast Software Encryption – FSE 1996*, volume 1039 of *Lecture Notes in Computer Science*, pages 71–82. Springer-Verlag, 1996.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 542–552. ACM Press, 1991.

[DDN95] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. Technical Report CS95-27, Weizmann Institute of Science, 1995.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

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

[DKXY03] Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung. Strong key-insulated signature schemes. In Y. Desmedt, editor, *Advances in Cryptology – Public-Key Cryptography 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 130–144. Springer-Verlag, January 2003.

[El 84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G.R. Blakely and D. Chaum, editors, *Advances in Cryptology – CRYPTO 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer-Verlag, 1984.

[FF02] Marc Fischlin and Roger Fischlin. The representation problem based on factoring. In B. Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 96–113. Springer-Verlag, February 2002.

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[FKK96] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The ssl protocol: Version 3.0. IETF Internet Draft, 1996.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. Odlyzko, editor, *Advances in Cryptology – CRYPTO 1986*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, August 1986.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

[GHR99] Rosario Gennaro, Shai Halevi, and Tal Rabin. Secure hash-and-sign signatures without the random oracle. In *Advances in Cryptology – EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 123–139. Springer-Verlag, 1999.

[Gir90] Marc Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number. In I. Damgård, editor, *Advances in Cryptology – EUROCRYPT 1990*, volume 473 of *Lecture Notes in Computer Science*, pages 481–486. Springer-Verlag, May 1990.

[Gir91] Marc Girault. Self-certified public keys. In D. Davies, editor, *Advances in Cryptology – EUROCRYPT 1991*, volume 547 of *Lecture Notes in Computer Science*, pages 490–497. Springer-Verlag, April 1991.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science*, pages 102–113. IEEE Computer Society Press, 2003.

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

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

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[Gol01] Oded Goldreich. *Foundations of Cryptography, Basic Tools*. Cambridge University Press, June 2001.

[GQ89] Louis C. Guillou and Jean-Jacques Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO 1988*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer-Verlag, August 1989.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In Y. Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer-Verlag, 2002.

[Gün90] Christoph G. Günther. An identity-based key-exchange protocol. In J-J. Quisquater and J. Vandewille, editors, *Advances in Cryptology – EUROCRYPT 1989*, volume 434 of *Lecture Notes in Computer Science*, pages 29–37. Springer-Verlag, 1990.

[GUQ01] Louis C. Guillou, Michel Ugon, and Jean-Jacques Quisquater. Cryptographic authentication protocols for smart cards. *Computer Networks*, 36(4):437–451, 2001.

[Hes03] Florian Hess. Efficient identity based signature schemes based on pairings. In K. Nyberg and H. Heys, editors, *Selected Areas in Cryptography, SAC 2002*, pages 310–324. Springer-Verlag, February 2003.

[HNZI99] Goichiro Hanaoka, Tsuyoshi Nishioka, Yuliang Zheng, and Hideki Imai. An efficient hierarchical identity-based key-sharing method resistant against collusion-attacks. In K. Lam, E. Okamoto, and C. Xing, editors, *Advances in Cryptology – ASIACRYPT 1999*, volume 1716 of *Lecture Notes in Computer Science*, pages 348–362. Springer-Verlag, 1999.

[Hoh03] Susan Hohenberger. The cryptographic impact of groups with infeasible inversion. Master's thesis, Massachusetts Institute of Technology, 2003.

[JMSW02] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In B. Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 244–262. Springer-Verlag, 2002.

[JN03] Antoine Joux and Kim Nguyen. Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4):239–247, 2003.

[JY96] Markus Jakobsson and Moti Yung. Revokable and versatile electronic money (extended abstract). In *Proceedings of the 3rd Conference on Computer and Communications Security*, pages 76–87. ACM Press, 1996.

[Kah96] David Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, revised edition, 1996.

[KH04] Kaoru Kurosawa and Swee-Huay Heng. From digital signature to ID-based identification/signature. In J. Zhou F. Bao, R. Deng, editor, *Advances in Cryptology – Public-Key Cryptography 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag, 2004.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO 1999*, Lecture Notes in Computer Science, pages 388–397. Springer-Verlag, 1999.

[KN93] John Kohl and B. Clifford Neuman. The Kerberos network authentication service (v5). Internet Request for Comments 1510, 1993.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO 1996*, Lecture Notes in Computer Science, pages 104–113. Springer-Verlag, 1996.

[LL93] Arjen K. Lenstra and Hendrik W. Lenstra Jr., editors. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.

[LMRS] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin, and Hovav Shacham. Sequential aggregate signatures from trapdoor permutations. Manuscript, available from `http://hovav.net/`.

[Low96] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In Tiziana Margaria and Bernhard Steffen, editors, *Tools and Algorithms for Construction and Analysis of Systems, Second International Workshop, TACAS '96*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer-Verlag, 1996.

[LV01] Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14(4):255–293, 2001.

[Mic94] Silvio Micali. A secure and efficient digital signature algorithm. Technical Report MIT/LCS/TM-501, Massachussets Institute of Technology, 1994.

[Mil86] Victor S. Miller. Use of elliptic curves in cryptography. In H. Williams, editor, *Advances in Cryptology – CRYPTO 1985*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer-Verlag, 1986.

[MR02a] Silvio Micali and Leonid Reyzin. Improving the exact security of digital signature schemes. *Journal of Cryptology*, 15(1):1–18, 2002.

[MR02b] Silvio Micali and Ronald Rivest. Transitive signature schemes. In B. Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 236–243. Springer-Verlag, 2002.

[MS88] Silvio Micali and Adi Shamir. An improvement of the Fiat–Shamir identification and signature scheme. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO 1988*, volume 403 of *Lecture Notes in Computer Science*. Springer-Verlag, 1988.

[MvOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[Nam02] Chanathip Namprempre. Secure channels based on authenticated encryption schemes: A simple characterization. In Y. Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[Nat77] National Bureau of Standards, U.S. Department of Commerce. Data encryption standard. Federal Information Processing Standard (FIPS) 46, 1977.

[Nat95] National Institute of Standards. Secure hash standard. Federal Information Processing Standards (FIPS) 180-1, 1995.

[NES03] NESSIE Consortium. Portfolio of recommended cryptographic primitives, 2003. Available from `http://www.cryptonessie.org`.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126. Springer-Verlag, 2002.

[NS78] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the Association for Computing Machinery*, 21(12):993–999, 1978.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen-ciphertext attacks. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*. ACM Press, 1990.

[Oka93] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In E. Brickell, editor, *Advances in Cryptology – CRYPTO 1992*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer-Verlag, August 1993.

[OO90] Kazuo Ohta and Tatsuaki Okamoto. A modification of the Fiat-Shamir scheme. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO 1988*, volume 403 of *Lecture Notes in Computer Science*, pages 232–243. Springer-Verlag, August 1990.

[OO98] Kazuo Ohta and Tatsuaki Okamoto. On concrete security treatment of signatures derived from identification. In H. Krawczyk, editor, *Advances in Cryptology – CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 354–370. Springer-Verlag, August 1998.

[OS90] H. Ong and Claus-Peter Schnorr. Fast signature generation with a Fiat-Shamir–like scheme. In I. Damgård, editor, *Advances in Cryptology – EUROCRYPT 1990*, volume 473 of *Lecture Notes in Computer Science*, pages 432–440. Springer-Verlag, May 1990.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology – EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 1999.

[Pat02] Kenneth G. Paterson. ID-based signatures from pairings on elliptic curves. Technical Report 2002/004, IACR ePrint Archive, January 2002.

[Pol78] John M. Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of Computation*, 32:918–924, 1978.

[PS98] Guillaume Poupard and Jacques Stern. Security analysis of a practical "on the fly" authentication and signature generation. In K. Nyberg, editor, *Advances in Cryptology – EUROCRYPT 1998*, volume 1403 of *Lecture Notes in Computer Science*, pages 422–436. Springer-Verlag, 1998.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[QQQ+90] Jean-Jacques Quisquater, Myriam Quisquater, Muriel Quisquater, Michaël Quisquater, Louis C. Guillou, Marie Annick Guillou, Gaïd Guillou, Anna Guillou, Gwenolé Guillou, Soazig Guillou, and Thomas A. Berson. How to explain zero-knowledge protocols to your children. In G. Brassard, editor, *Advances in Cryptology – CRYPTO 1989*, volume 435 of *Lecture Notes in Computer Science*, pages 628–631. Springer-Verlag, 1990.

[Rab79] Michael O. Rabin. Digital signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachussets Institute of Technology, 1979.

[Riv00] Ronald Rivest. Two signature schemes. Slides from talk given at Cambridge University, October 17, 2000, 2000.

[RS92] Charles Rackoff and Daniel Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *Advances in Cryptology – CRYPTO 1991*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer-Verlag, 1992.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, 1978.

[SBZ02] Ron Steinfeld, Laurence Bull, and Yuliang Zheng. Content extraction signatures. In Kwangjo Kim, editor, *Information Security and Cryptology - ICISC 2001*, volume 2288 of *Lecture Notes in Computer Science*, pages 285–304. Springer-Verlag, 2002.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smartcards. In G. Brassard, editor, *Advances in Cryptology – CRYPTO 1989*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer-Verlag, August 1990.

[Sch96] Claus-Peter Schnorr. Security of 2^t-root identification and signatures. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO 1996*, volume 1109 of *Lecture Notes in Computer Science*, pages 143–156. Springer-Verlag, August 1996.

[Sch99] Berry Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In M. Wiener, editor, *Advances in Cryptology – CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 148–164. Springer-Verlag, 1999.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G.R. Blakely and D. Chaum, editors, *Advances in Cryptology – CRYPTO 1984*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.

[Sho99] Victor Shoup. On the security of a practical identification scheme. *Journal of Cryptology*, 12(4):247–260, 1999.

[Sho02] Victor Shoup. OAEP reconsidered. *Journal of Cryptology*, 15(4):223–249, 2002.

[SOK00] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *SCIS 2000*, Okinawa, Japan, January 2000.

[SPMLS02] Jacques Stern, David Pointcheval, John Malone-Lee, and Nigel P. Smart. Flaws in applying proof methodologies to signature schemes. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 93–110. Springer-Verlag, August 2002.

[SR98] Bruce Schneier and James Riordan. A certified e-mail protocol. In *14th Annual Computer Security Applications Conference (ACSAC'98)*, pages 347–352. IEEE Computer Society, 1998.

[SSN98] Shahrokh Saeednia and Reihaneh Safavi-Naini. On the security of Girault's identification scheme. In H. Imai and Y. Zheng, editors, *Advances in Cryptology – Public-Key Cryptography 1998*, volume 1431 of *Lecture Notes in Computer Science*, pages 149–153. Springer-Verlag, 1998.

[ST03] Adi Shamir and Eran Tromer. Factoring large numbers with the TWIRL device. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 1–26. Springer-Verlag, 2003.

[Yi03] Xun Yi. An identity-based signature scheme from the Weil pairing. *IEEE Communications Letters*, 7(2):76–78, 2003.

Publications

[1] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[2] Bart De Decker, Gregory Neven, and Frank Piessens. Secure Vickrey auctions without a trusted third party. In D. Gritzalis, S. de Capitani di Vimercati, P. Samarati and S. Katsikas editors, *Security and Privacy in the Age of Uncertainty, IFIP TC11 18th International Conference on Information Security (SEC2003)*, volume 250 of *IFIP Conference Proceedings*, pages 337–348. Kluwer Academic Publishers, 2003.

[3] Mihir Bellare and Gregory Neven. Transitive signatures based on factoring and RSA. In Y. Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 397–414. Springer-Verlag, 2002.

[4] Bart De Decker, Gregory Neven, Frank Piessens and Erik Van Hoeymissen. Second price auctions: a case study of secure distributed computing. In K. Zielinski, K. Geihs, A. Laurentowski editors, *New Developments in Distributed Applications and Interoperable Systems, IFIP TC6 / WG6.1 Third International Working Conference on Distributed Applications and Interoperable Systems*, volume 198 of *IFIP Conference Proceedings*, pages 217–228. Kluwer Academic Publishers, 2001.

[5] Gregory Neven, Erik Van Hoeymissen, Bart De Decker, and Frank Piessens. Enabling secure distributed computations: semi-trusted hosts and mobile agents. *Networking and Information Systems Journal*, 3:1–18, 2000.

[6] Thomas Herlea, Joris Claessens, Gregory Neven, Frank Piessens, Bart Preneel, and Bart De Decker. On securely scheduling a meeting. In M. Dupuy and P. Paradinas, editors, *Trusted Information: The New Decade Challenge, IFIP TC11 Sixteenth Annual Working Conference on Information Security*

(IFIP/Sec'01), volume 193 of *IFIP Conference Proceedings*. Kluwer Academic Publishers, 2001.

[7] Stijn Van den Enden, Erik Van Hoeymissen, Gregory Neven, and Pierre Verbaeten. A case study in application integration. At *OOPSLA Business Objects and Components Design and Implementation Workshop VI: Enterprise Application Integration*, October 2000, Minneapolis, Minnesota (USA).

[8] Frank Piessens, Bart De Decker, Erik Van Hoeymissen and Gregory Neven. On the trade-off between communication and trust in secure computations. At *ECOOP Workshop on Mobile Object Systems*, Cannes, France, June 13, 2000.

[9] Bart De Decker, Frank Piessens, Erik Van Hoeymissen, and Gregory Neven. Semi-trusted hosts and mobile agents: enabling secure distributed computations. In E. Horlait, editor, *Mobile Agents for Telecommunication Applications, Second International Workshop, MATA 2000*, volume 1931 of *Lecture Notes in Computer Science*, pages 219–232. Springer, 2000.

[10] Gregory Neven, Frank Piessens, and Bart De Decker. On the practical feasibility of secure distributed computing: a case study. In S. Qing and J. Eloff, editors, *Information Security for Global Information Infrastructures, IFIP TC11 Fifteenth Annual Working Conference on Information Security*, volume 175 of *IFIP Conference Proceedings*, pages 361–370. Kluwer Academic Publishers, 2000.

Biography

Gregory Neven was born on January 28, 1978 in Maasmechelen, Belgium. He received a Bachelor's degree of Science in Engineering (Kandidaat Burgerlijk Ingenieur) and a Master's degree of Science in Engineering in Computer Science (Burgerlijk Ingenieur in de Computerwetenschappen) from the Katholieke Universiteit Leuven in Belgium. He graduated summa cum laude in July 2000 with the thesis "Secure Distributed Computing", supervised by Frank Piessens. This work was awarded with the IBM Belgium Awards for Informatics in 2001, and won a third prize at the KVIV Thesis Awards in 2000. He started working as a Ph.D student at the DistriNet working group of the Department of Computer Science at K.U.Leuven in October 2000, funded by a research assistantship (aspirantenmandaat) from the Fund for Scientific Research – Flanders (F.W.O.–Vlaanderen). He was a visiting scholar at the University of California at San Diego (UCSD) working with Mihir Bellare from January until September 2002, and again from July until October 2003.

Bewijsbaar Veilige Identiteitsgebaseerde Identificatieschema's en Transitieve Handtekeningen

Gregory NEVEN

Beknopte Samenvatting

Tot de jaren tachtig was cryptografisch ontwerp eerder een vaardigheid dan een echte wetenschap: schema's werden voorgesteld met hooguit een intuïtie voor hun veiligheid, met als enig criterium de jarenlange weerstand tegen aanvallen van experts. Een modernere aanpak is *bewijsbare veiligheid*. Volgens deze aanpak beschrijft de ontwerper van een schema eerst duidelijk wat verstaan wordt onder de veiligheid van het schema. Vervolgens toont hij wiskundig aan dat het schema enkel kan gebroken worden door een onveilig onderliggend cryptografisch bouwblok te breken, of door een wiskundige doorbraak te realiseren. Bewijsbare veiligheid is geëvolueerd van een speeltje voor theoretici tot een belangrijke schema-eigenschap waarmee rekening wordt gehouden bij de keuze van industriestandaarden.

In deze thesis bestuderen we de bewijsbare veiligheid van enkele geselecteerde cryptografische primitieven. We stellen eerst bruikbare maar haalbare veiligheidsnoties op, en bewijzen vervolgens de veiligheid van bestaande en nieuwe schema's onder deze noties.

Het eerste deel behandelt *identiteitsgebaseerde identificatie-* en *handtekeningsschema's*. Dit zijn cryptografische primitieven voor entiteits- en boodschapsauthenticatie, respectievelijk, waarbij de publieke sleutel van een gebruiker eenvoudigweg zijn identiteit is (in plaats van een willekeurig getal dat op een veilige manier aan de gebruiker moet worden toegekend). We stellen eerst een raamwerk van veiligheidsbewarende transformaties voor. Vervolgens gebruiken we dit raamwerk om de veiligheid te bewijzen (en in een enkel geval, te breken) van schema's van 13 verschillende "families" die in de loop van de afgelopen twee decennia voorgesteld werden, maar waarvoor nog geen bewijs gekend was.

In het tweede deel bespreken we *transitieve handtekeningsschema's*. Dit zijn handtekeningsschema's die toelaten verbindingen in een grafe te ondertekenen zodat elke gebruiker (en dus niet enkel de ondertekenaar) vanuit twee handtekeningen voor aanliggende verbindingen $\{i, j\}$ en $\{j, k\}$ een derde handtekening kan berekenen voor de rechtstreekse verbinding $\{i, k\}$. We beantwoorden een open vraag betreffende de veiligheid van een bestaand schema, en stellen nieuwe, bewijsbaar veilige schema's voor die efficiëntievoordelen bieden ten opzichte van bestaande schema's.

1 Inleiding

Cryptografie is een oude vaardigheid, maar is relatief jong als een echte wetenschap. Vierduizend jaar geleden al weken de oude Egyptenaren regelmatig af van de gangbare hiëroglyfische notatie om de inhoud belangrijker te doen lijken. De Spartanen schreven geheime boodschappen langs de as van een stok waarrond een strook perkament gewikkeld was; het perkament werd afgerold getransporteerd, en de ontvanger reconstrueerde de boodschap door het weer op te rollen rond een stok van dezelfde diameter. Julius Caesar gebruikte een naar hem genoemde cijfer die elke letter vervangt door de letter drie plaatsen verder in het alfabet.

Dergelijke technieken boden misschien een redelijke bescherming in een maatschappij met een grotendeels ongeletterde bevolking, maar schieten duidelijk tekort voor de huidige communicatienoden. Met het Internet als globale informatie-infrastructuur zijn gegevens die vroeger slechts mits een aanzienlijke inspanning konden verkregen worden, tegenwoordig via enkele muisklikken beschikbaar. De keerzijde van de medaille is een verhoogde blootstelling aan kwaadaardige gebruikers, die bovendien aangetrokken worden door een hogere verwachte buit. Met een wettelijke bescherming die te traag en ingewikkeld blijkt om als afdoend afschrikkingsmiddel dienst te doen, is het voorkomen van aanvallen door technische middelen belangrijker dan ooit.

Niet toevallig nam de publieke aandacht voor cryptografie een hoge vlucht rond dezelfde tijd dat digitale informatie haar plaats begon op te eisen in de maatschappij. Tijdens de jaren zeventig werd de blokcijfer DES (Data Encryption Standard [Nat77]) ontwikkeld, maar de echte doorbraak kwam met de conceptuele uitvinding van publieke-sleutel cryptografie door Diffie en Hellman in 1976 [DH76]. In de plaats van Alice en Bob op voorhand een geheime sleutel te laten afspreken om later hun gesprekken mee te beveiligen, stelden Diffie en Hellman voor sleutels in *paren* te laten voorkomen, bestaande uit een *publieke* en een *private* sleutel. Alice publiceert haar publieke sleutel, maar houdt haar private sleutel strikt geheim. Als Bob een boodschap naar Alice wil sturen, encrypteert hij die met Alice's publieke sleutel; de resulterende *cijfertekst* kan enkel met Alice's private sleutel terug gedecrypteerd worden. Natuurlijk moet er een wiskundig verband bestaan tussen de publieke en de private sleutel, maar het schema wordt zodanig ontworpen dat het onmogelijk is binnen een "redelijke tijd" de private sleutel te berekenen horende bij een bepaalde publieke sleutel. Twee jaar later publiceerden Rivest, Shamir en Adleman [RSA78] het eerste concrete publieke-sleutel encryptieschema, het *RSA* algoritme, een bijdrage die in 2002 met de ACM Turing Award bekroond werd.

Bewijsbare veiligheid. Alhoewel de uitvinding van publieke-sleutel cryptografie meteen de aandacht trok van talrijke goede wetenschappers, bleef het ontwerp van cryptografische protocols en primitieven gedurende minstens een

decennium eerder een vaardigheid dan een echte wetenschap. Schema's werden voorgesteld met hooguit een intuïtie waarom ze moeilijk te breken zouden zijn, met als enige maatstaf voor veiligheid de jarenlange weerstand van het schema tegen aanvallen van experts in het onderzoeksdomein.

In het begin van de jaren 1980 introduceerden Goldwasser en Micali [GM84] een nieuwe aanpak: *bewijsbare veiligheid*, soms toepasselijker *reductionistische veiligheid* genoemd. Het idee van bewijsbare veiligheid is wiskundig aan te tonen dat elke succesvolle aanval op een schema kan omgevormd worden in een aanval op een onderliggend primitief, of in de oplossing van een wiskundig probleem waarvan algemeen wordt aangenomen dat het niet "efficiënt" (d.w.z. in polynomiale tijd) kan opgelost worden. Dit verbindt de veiligheid van een schema rechtstreeks aan de veiligheid van de onderliggende bouwblokken, en reduceert de opties van een aanvaller tot het breken van een onveilige subcomponent of tot het realiseren van een wiskundige doorbraak.

Sindsdien is bewijsbare veiligheid uitgegroeid van een speeltje voor theoretici tot een belangrijke eigenschap van een schema, die ook in rekening gebracht wordt bij het vastleggen van industriestandaarden.

Het willekeurig-orakelmodel. Aangemoedigd door het gebrek aan efficiënte bewijsbaar veilige cryptografische constructies om de heuristische schema's te vervangen die in het midden van de jaren negentig in gebruik waren, stelden Bellare en Rogaway [BR93a, Bel98] het *willekeurig-orakelmodel* voor als compromis tussen theorie en praktijk. Het idee is de veiligheid te bewijzen in een "denkbeeldige wereld" waarin alle deelnemers, zowel goede als slechte, toegang hebben tot een orakel $\mathrm{H}(\cdot)$ dat een willekeurige functie implementeert. In de praktijk wordt dit willekeurig orakel vervangen door een cryptografische hash-functie als SHA-1 [Nat95] of RIPEMD-160 [DBP96], in de hoop dat deze het onvoorspelbaar gedrag van het willekeurige orakel voldoende imiteert om de veiligheid te blijven waarborgen in de "echte wereld".

Strikt gezien is een bewijs in het willekeurig-orakelmodel niet langer een bewijs, maar hoogstens nog een goede heuristiek, want (berekenbare) hash-functies kunnen vanzelfsprekend nooit onvoorspelbaar zijn. Er bestaat dan ook heel wat controverse rond het willekeurig-orakelmodel in de cryptografische onderzoeksgemeenschap [CGH98, Nie02, GK03], en inderdaad, bewijzen in het willekeurig-orakelmodel moeten met enige voorzichtigheid benaderd worden. Anderzijds is het een zeer waardevol hulpmiddel gebleken voor het bestuderen van de veiligheid van efficiënte schema's die jarenlang zowel aanval als bewijs weerstonden. Bovendien heeft een schema met een bewijs in het willekeurig-orakelmodel nog steeds een sterke voorkeur boven volledig ad-hoc protocolontwerp. Het is met deze waarschuwing in het achterhoofd dat ook dit werk gebruik maakt van willekeurige orakels.

Onderwerp van deze thesis. Dit werk past de aanpak van bewijsbare veiligheid toe op een aantal geselecteerde cryptografische primitieven door eerst een

nuttige en haalbare veiligheidsnotie te definiëren, en vervolgens veiligheid onder die notie aan te tonen, zowel voor bestaande maar onbewezen schema's als voor volledig nieuwe schema's.

Deze thesis bestaat uit twee delen. In het eerste deel behandelen we bewijsbaar veilige *identiteitsgebaseerde identificatie- en handtekeningsschema's*. Identificatieschema's zijn cryptografische primitieven om *entiteitsauthenticatie* te verzorgen, d.w.z. ze bieden een garantie dat entiteiten (gebruikers, computers, ...) zijn wie ze beweren te zijn. Handtekeningsschema's zijn primitieven voor *boodschapsauthenticatie*, en garanderen dat boodschappen afkomstig zijn van de zender van wie ze lijken afkomstig te zijn. *Identiteitsgebaseerde* [Sha84] primitieven laten toe de identiteit of het email-adres van een gebruiker als publieke sleutel te gebruiken, hetgeen efficiëntievoordelen biedt in vergelijking met de klassieke techniek van certificaten om publieke sleutels aan gebruikers te verbinden. In dit werk stellen we een veiligheidsbewarende transformatie voor die we vervolgens toepassen om de veiligheid te bewijzen van een dozijn identiteitsgebaseerde identificatie- en handtekeningsschema's die in de voorbije twee decennia voorgesteld werden.

Het tweede deel van deze thesis handelt over transitieve handtekeningen [MR02b]. Dit zijn digitale handtekeningen die toelaten verbindingen in een grafe te ondertekenen zodat eender wie vanuit twee handtekeningen voor aanliggende verbindingen $\{i, j\}$ en $\{j, k\}$ een geldig derde handtekening kan berekenen voor de rechtstreekse verbinding $\{i, k\}$. We beantwoorden een open vraag gesteld door [MR02b], en presenteren een aantal nieuwe schema's die aanzienlijke efficiëntievoordelen bieden ten opzichte van de bestaande schema's.

Overzicht. In Sectie 2 introduceren we de notatie die doorheen de tekst zal gehanteerd worden, gaan we iets dieper in op bewijsbare veiligheid, en beschrijven we kort de wiskundige problemen waarop de schema's in deze tekst gebaseerd zijn. Sectie 3 vat onze resultaten samen voor identiteitsgebaseerde identificatie- en handtekeningsschema's, en Sectie 4 doet hetzelfde voor transitieve handtekeningen. We besluiten de thesis in Sectie 5 met enkele suggesties voor verder onderzoek.

2 Voorkennis

Notatie. Definieer $\{0, 1\}$ als de verzameling van individuele bits, en $\{0, 1\}^*$ als de verzameling van alle bitstrings. Laat $\mathbb{N} = \{0, 1, 2, \ldots\}$ de verzameling van natuurlijke getallen voorstellen. Als $k \in \mathbb{N}$, dan is 1^k de string van k één-symbolen en is $\{0, 1\}^k$ de verzameling van bitstrings van lengte k. De lege bitstring wordt voorgesteld door ε. Als x, y strings zijn, dan is $|x|$ de lengte van x en is $x\|y$ de concatenatie van x en y. Als S een verzameling is, dan is $|S|$ de cardinaliteit van S. Met de notatie $x \xleftarrow{R} S$ bedoelen we dat een element x geselecteerd wordt uit

S volgens een uniforme verdeling. Een functie $f : \mathbb{N} \to [0,1]$ is *verwaarloosbaar* als hij sneller naar nul gaat dan de inverse van eender welke veelterm, oftewel voor elke exponent $c \in \mathbb{N}$ bestaat er een $k_c \in \mathbb{N}$ zodat $f(k) \leq k^{-c}$ voor alle $k > k_c$.

Als A een (mogelijk probabilistisch) algoritme is met toegang tot orakels $\textsc{Or}_1, \ldots, \textsc{Or}_m$, dan betekent $y \xleftarrow{R} \mathsf{A}(x_1, \ldots, x_n : \textsc{Or}_1, \ldots, \textsc{Or}_m)$ dat het resultaat van een uitvoering van A op invoer $x_1, \ldots, x_n$ wordt toegekend aan de variabele y. Een interactief algoritme is een toestandshebbend algoritme dat op invoer een inkomend bericht M_{in} en een toestand St, een uitgaand bericht M_{out} en een vernieuwde toestand St' als uitvoer teruggeeft; dit noteren we als $(M_{\text{out}}, St') \xleftarrow{R} \mathsf{A}(M_{\text{in}}, St : \textsc{Or}_1, \ldots, \textsc{Or}_m)$. Een algoritme is *polynomiale-tijd* als de uitvoeringstijd ervan begrensd is door een veelterm in de lengte van de invoer.

Bewijsbare veiligheid. Vooraleer zinvolle uitspraken te kunnen doen over de veiligheid van een cryptografisch schema, moet het eerst perfect duidelijk zijn wat er onder "veiligheid" verstaan wordt. Dit wordt vastgelegd door de *veiligheidsnotie* geassocieerd aan het primitief. Dit is een spel of experiment waarin een tegenstander, gemodelleerd als een algoritme, uitgedaagd wordt om met behulp van bepaalde inputs en orakels een aanval te plegen op het schema.

Het veilgheidsexperiment is geparametriseerd met een veiligheidsparameter k, typisch de lengte van een sleutel gebruikt in het experiment. We modelleren de tegenstander als een algoritme A, en definiëren het *voordeel* $\mathbf{Adv}^{\text{sec}}_{\mathcal{S},\mathsf{A}}(k)$ van een tegenstander A in het aanvallen van schema $\mathcal{S}$ onder veiligheidsnotie sec als de kans dat A het spel wint. We zeggen dat het schema *veilig* is onder notie sec als er geen polynomiale-tijd tegenstander A bestaat met een niet-verwaarloosbaar voordeel in het breken van $\mathcal{S}$. Het veiligheidsbewijs toont aan dat een dergelijke tegenstander inderdaad niet kan bestaan, meestal door een bewijs uit het ongerijmde: veronderstel dat deze tegenstander wel bestaat, dan bestaat er tevens een algoritme dat hetzij een aanval pleegt op een onderliggend primitief, hetzij een wiskundig probleem oplost waarvan algemeen wordt aangenomen dat er geen efficiënte oplossingen voor bestaan.

Moeilijke wiskundige problemen. De meeste schema's behandeld in deze tekst zijn bewijsbaar veilig onder de veronderstelling dat een bepaald wiskundig probleem niet efficiënt oplosbaar is. We bespreken hier kort enkele voorbeelden van zulke problemen.

Veruit het meest gekende (en meest bestudeerde) probleem is dat van het factoriseren van grote getallen. Ondanks aanzienlijke inspanningen van wiskundigen over de hele wereld, bestaat er geen polynomiale-tijd algoritme dat, gegeven een getal $N = pq$ waarbij p en q twee grote priemgetallen zijn, de factoren p en q berekent.

Het *RSA-probleem* [RSA78] is nauw verwant aan het factorisatieprobleem: gegeven een modulus N zoals hierboven, een exponent e zodat $\gcd(e, \varphi(N)) = 1$

en een willekeurig element $y \in \mathbb{Z}_N^*$, bereken $x \in \mathbb{Z}_N^*$ zodat $x^e \equiv y \bmod N$. Hierbij is $\mathbb{Z}_N^*$ de multiplicatieve groep modulo N, en is $\varphi(N) = (p-1)(q-1)$ de orde ervan. Het RSA-probleem is gemakkelijk op te lossen als N kan gefactoriseerd worden, maar het is onbekend of het omgekeerde ook waar is.

Een recente variant op het RSA-probleem is het *één-meer RSA-probleem* [BNPS03], waarbij de tegenstander een modulus N en exponent e als invoer krijgt, en toegang heeft tot twee orakels. Het eerste is een uitdagingsorakel dat op elke invocatie een nieuwe uitdaging y_i genereert, willekeurig gekozen uit $\mathbb{Z}_N^*$. Het tweede is een inversie-orakel, dat op invoer $y \in \mathbb{Z}_N^*$ antwoordt met $x \in \mathbb{Z}_N^*$ zodat $x^e \equiv y \bmod N$. De taak van de tegenstander bestaat erin *alle* uitdagingen van het uitdagingsorakel te inverteren met behulp van een aantal bevragingen van het inversie-orakel dat strikt kleiner is dan het aantal geïnverteerde uitdagingen. Het is duidelijk dat dit probleem gemakkelijk oplosbaar is als het RSA-probleem dat ook is; de veronderstelling dat het één-meer RSA-probleem moeilijk op te lossen is, is dus zwaarder dan de gewone RSA-veronderstelling.

Een andere klasse van getaltheoretische problemen die gebruikt worden in de cryptografie zijn gebaseerd op *discrete logaritmen*. In een multiplicatieve groep $\mathbb{G}$ van orde q met generator g is het discrete logaritme van $y \in \mathbb{G}$ ten opzichte van g gedefinieerd als het unieke getal $x \in \mathbb{Z}_q$ zodat $g^x \equiv y$. Voorbeelden van groepen waarin het berekenen van discrete logaritmen moeilijk verondersteld wordt zijn primale-orde subgroepen van $\mathbb{Z}_p^*$, met p een priemgetal, en elliptische curven. Net als bij het RSA-probleem kunnen we het *één-meer discrete-logaritmeprobleem* [BNPS03] definiëren waarbij de tegenstander toegang krijgt tot een uitdagingsorakel en een discrete-logaritme-orakel, en hij het discrete logaritme van alle uitdagingen moet berekenen met strikt minder bevragingen van het discrete-logaritme-orakel.

Twee problemen gerelateerd aan discete logaritmen zijn het *computationele Diffie-Hellman (CDH)* [DH76] en het *beslissing Diffie-Hellman (DDH)* probleem. Het CDH-probleem is, gegeven $u \equiv g^a$ en $v \equiv g^b$, het element w te berekenen zodat $w \equiv g^{ab}$; het DDH-probleem bestaat erin voor een gegeven $u \equiv g^a$, $v \equiv g^b$ en w te beslissen of $w \equiv g^{ab}$ of niet. Beide problemen zijn gemakkelijk als het berekenen van discrete logaritmen gemakkelijk is, en het DDH-probleem is gemakkelijk als het CDH-probleem gemakkelijk is. Implicaties in de omgekeerde richting zijn opnieuw onbekend, maar recentelijk werden de zogenaamde *paring-groepen* ontdekt (op basis van de Weil en Tate paringsfuncties over supersinguliere elliptische curven; zie [JN03, BF01] voor meer details) waarin het DDH-probleem gemakkelijk is, maar waarin het CDH-probleem nog steeds verondersteld wordt moeilijk oplosbaar te zijn. Groepen met deze eigenschap worden ook wel algemener *kloof-Diffie-Hellmangroepen (GDH-groepen)* genoemd, maar voorlopig zijn de paring-groepen het enige bekende voorbeeld ervan. Ook van het CDH-probleem bestaat een één-meer variant, het zogenaamde *één-meer CDH-probleem* [Bol03a].

3 Identiteitsgebaseerde Identificatieschema's

3.1 Achtergrond

Tijdens het einde van de jaren tachtig en het begin van de jaren negentig werden talrijke identiteitsgebaseerde identificatieschema's (IBI-schema's) en identiteitsgebaseerde handtekeningsschema's (IBS-schema's) voorgesteld, zoals onder meer de Fiat-Shamir IBI- en IBS-schema's [FS86], het IBS-schema in de paper van Shamir [Sha84] waarmee hij het concept van identiteitsgebaseerde cryptografie introduceerde, en andere schema's [Oka93, Gir90, Bet88]. Meer recentelijk werden ook een aantal paring-gebaseerde IBS-schema's voorgesteld [SOK00, Hes03, Pat02, CC03, Yi03].

Alhoewel er heel wat literatuur bestaat over de bewijsbare veiligheid van identificatieschema's, beperkt dit werk zich tot standaard identificatieschema's (SI schema's), en houdt het geen rekening met de bijkomende risico's geïntroduceerd door het identiteitsgebaseerde aspect. Zo bestaan er bijvoorbeeld veiligheidsbewijzen voor SI-schema's die nauw verwant zijn met de IBI-schema's van Fiat-Shamir en Guillou-Quisquater [FS86, GQ89], maar niet voor de IBI-schema's zelf. Sterker nog, een bewijsbaar veilige aanpak voor IBI-schema's ontbreekt volkomen: er zijn geen geschikte veiligheidsnoties, en bijgevolg werd ook geen van de voorgestelde IBI-schema's veilig bewezen.

De situatie voor IBS-schema's is iets beter. Cha en Cheon geven een degelijke veiligheidsdefinitie voor IBS-schema's en bewijzen de veiligheid van hun paring-gebaseerd schema [CC03]. Dodis, Katz, Xu en Yung [DKXY03] definiëren een klasse van standaard handtekeningsschema's (SH-schema's) die ze *valdeur* SH-schema's noemen, en presenteren een algemene transformatie die elk veilig valdeur SH-schema omzet in een veilig IBS-schema. De bewijsbare veiligheid van verscheidene bestaande IBS-schema's volgt uit het toepassen van deze transformatie op SH-schema's waarvan de veiligheid eerder al bewezen werd. Desalniettemin blijft de veiligheid van verscheidene IBS-schema's onbewezen, hetzij omdat ze niet het resultaat zijn van de transformatie toegepast op een valdeur SH-schema, hetzij omdat de veiligheid van het onderliggend SH-schema nooit geanalyseerd werd.

Dit werk vult de bovenstaande leemtes betreffende IBI- en IBS-schema's op. De eerste stap is het opstellen van geschikte veiligheidsdefinitie voor IBI-schema's. Vervolgens presenteren we een raamwerk van nieuwe en bestaande veiligheidsbewarende transformaties zoals afgebeeld in Figuur 5.1. Dit raamwerk reduceert het veilig bewijzen van IBI- en IBS-schema's tot het aantonen van de veiligheid van een onderliggend SI-schema, hetgeen een aanzienlijk eenvoudigere taak is. Met dit raamwerk als gereedschap in de hand analyseren we de veiligheid van twee decennia aan voorgestelde IBI- en IBS-schema's onder de vorm van 13 *families* van schema's, en behalen nieuwe resultaten hetzij door toepassing van onze transformaties op bewijsbaar veilige schema's, hetzij door het

bewijzen van voorheen niet-geanalyseerde onderliggende schema's, hetzij door het voorgestelde schema te breken. Tenslotte bespreken we twee uitzonderlijke IBI-schema's (waarvan één gekend en één nieuw) die, alhoewel ze niet het resultaat zijn van het toepassen van onze transformaties, wel rechtstreeks als IBI-schema's veilig kunnen bewezen worden. Voor zover we weten zijn dit de enige IBI en IBS-schema's bekend in de literatuur die niet omvat worden door ons raamwerk. Figuur 5.2 vat onze resultaten voor specifieke schema's samen.

3.2 Definities en Veiligheidsnoties

Standaard identificatieschema's. Een *standaard identificatieschema (SI-schema)* is een tupel $\mathcal{SI} = (\mathsf{Kg}, \mathsf{P}, \mathsf{V})$ van 3 algoritmes, waar Kg een gerandomiseerd polynomiale-tijd sleutelgeneratie-algoritme is, en waar P en V polynomiale-tijd interactieve algoritmes zijn genaamd de *bewijzer* en de *verifieerder*. Initieel voert de bewijzer $\mathsf{Kg}(1^k)$ uit, waar $k \in \mathbb{N}$ de veiligheidsparameter is, om een sleutelpaar (pk, sk) te verkrijgen. Hij publiceert de publieke sleutel pk, maar houdt de private sleutel sk geheim. Tijdens het interactieve identificatieprotocol voert de bewijzer P uit met sk als initiële toestand, en voert de verifieerder V uit met pk als initiële toestand. Het protocol eindigt wanneer V in de `acc` of `rej` toestand terechtkomt, waarmee hij te kennen geeft dat hij de conversatie accepteert, respectievelijk verwerpt.

De veiligheidsnotie die we beogen is die van impersonificatie onder passieve, actieve en concurrente aanval. Een SI-tegenstander A wordt voorgesteld door een koppel algoritmes $(\mathsf{CV}, \mathsf{CP})$, waarbij CV de *valse verifieerder* wordt genoemd en CP de *valse bewijzer*. De aanval verloopt in twee fasen. De eerste is een leerfase, waarin de valse verifieerder een verse publieke sleutel als invoer krijgt, en waarin hij bovendien toegang heeft tot een orakel dat ofwel afschriften genereert van geslaagde conversaties tussen een echte bewijzer en verifieerder (passieve aanval [FFS88]), ofwel de valse verifieerder toelaat te interageren met echte bewijzers geïnitialiseerd met de overeenkomstige private sleutel (actieve [FFS88] en concurrente [BP02] aanval). Bij een actieve aanval moet de valse verifieerder de sessies met verschillende bewijzers sequentieel afwerken, bij een concurrente aanval mogen de verschillende sessies op een willekeurige manier met mekaar verweven worden. De eerste fase eindigt als CV zijn uitvoer beëindigt en toestandsinformatie St als uitvoer teruggeeft. Deze toestandsinformatie wordt in de tweede fase als invoer aan de valse bewijzer CP gegeven. De tweede fase is de zogenaamde impersonatiefase, waarin CP wordt geconfronteerd met een echte verifieerder V geïnitialiseerd met pk en moet trachten op basis van de toestandsinformatie St de verifieerder te doen accepteren, zonder enige hulp van bijkomende orakels. Het voordeel van tegenstander A om het SI-schema $\mathcal{SI}$ te breken onder passieve (pa), actieve (aa) en concurrente (ca) aanval wordt genoteerd als $\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{SI},\mathsf{A}}(k)$, atk $\in \{\text{pa}, \text{aa}, \text{ca}\}$, en is gedefinieerd als de kans van van

A om het bovenstaande spel te winnen. We zeggen dat het $\mathcal{SI}$ imp-atk veilig is als $\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{SI},\mathsf{A}}(k)$ een verwaarloosbare functie is in k voor alle polynomiale-tijd tegenstanders A.

Identiteitsgebaseerde identificatieschema's. We definiëren een *identiteitsgebaseerd identificatieschema (IBI-schema)* als een tupel $\mathcal{IBI} = (\mathsf{MKg}, \mathsf{UKg}, \overline{\mathsf{P}}, \overline{\mathsf{V}})$ van vier polynomiale-tijd algoritmes. Een vertrouwde autoriteit voert het *hoofdsleutelgeneratie-algoritme* MKg uit op invoer 1^k om een hoofdsleutelpaar (mpk, msk) te verkrijgen. De sleutelautoriteit publiceert de publieke hoofdsleutel mpk als een systeem-wijde parameter, en houdt de private hoofdsleutel msk geheim. Op aanvraag van een gebruiker met identiteit $I \in \{0,1\}^*$ voert hij het *gebruikerssleutelgeneratie-algoritme* uit met als invoer msk en I om de geheime gebruikerssleutel usk voor identiteit I te berekenen. (We veronderstellen dat de sleutelautoriteit de identiteit van de gebruiker controleert en hem de sleutel usk op een veilige manier bezorgt.) In het interactieve identificatieprotocol voert de bewijzer met identiteit I het algoritme $\overline{\mathsf{P}}$ uit met begintoestand usk, en de verifieerder voert het $\overline{\mathsf{V}}$ algoritme uit geïnitialiseerd met mpk, I, totdat de verifieerder in de `acc` of `rej` toestand terechtkomt.

Net als bij SI-schema's is een IBI-tegenstander $\overline{\mathsf{A}}$ een koppel algoritmes $(\overline{\mathsf{CV}}, \overline{\mathsf{CP}})$ en verloopt het veiligheidsexperiment in twee fasen waarbij we een onderscheid maken tussen passieve, actieve en concurrente aanvallen. Om de bijkomende risico's te modelleren die geïntroduceerd worden door het identiteitsgebaseerde aspect, zoals bijvoorbeeld aanvallen vanwege interne en mogelijk zelfs samenzwerende gebruikers van het systeem, krijgt de aanvaller toegang tot twee bijkomende orakels: een initialisatie-orakel, waaraan $\overline{\mathsf{CV}}$ middels een bevraging I te kennen geeft een gebruiker met identiteit I te willen initialiseren, en een corruptie-orakel, waarmee $\overline{\mathsf{CV}}$ de geheime gebruikerssleutel horende bij een geïnitialiseerde identiteit I kan opvragen. De taak van de aanvaller bestaat erin zich ten opzichte van de echte verifieerder succesvol voor te doen als een identiteit J naar die de aanvaller zelf mag bepalen, zolang J voorheen niet gecorrumpeerd werd. Het voordeel van $\overline{\mathsf{A}}$ is zijn kans om bovenstaand experiment te winnen en wordt genoteerd als $\mathbf{Adv}^{\text{imp-atk}}_{\mathcal{IBI},\overline{\mathsf{A}}}(k)$. Het IBI-schema $\mathcal{IBI}$ is imp-atk veilig als dit voordeel verwaarloosbaar is voor alle polynomiale-tijd tegenstanders $\overline{\mathsf{A}}$.

Standaard handtekeningsschema's. Een *standaard handtekeningsschema (SS-schema)* $\mathcal{SS}$ is een tupel van drie algoritmes $(\mathsf{Kg}, \mathsf{Sign}, \mathsf{Vf})$, waarbij het sleutelgeneratie-algoritme Kg op invoer 1^k een sleutelpaar (pk, sk) genereert. Het handtekeningsalgoritme Sign berekent op invoer de private sleutel sk en boodschap $M \in \{0,1\}^*$ een handtekening σ voor M. Op basis van de publieke sleutel pk, een boodschap M en een handtekening σ beslist het verificatie-algoritme Vf of het handtekening geldig is of niet.

De meest aanvaarde veiligheidsnotie voor SS-schema's is die van *existentiële*

onvervalsbaarheid onder gekozen-boodschap aanval (uf-cma), waarbij een tegenstander F de publieke sleutel als invoer krijgt, en toegang heeft tot een handtekeningsorakel waaraan hij handtekeningen kan opvragen voor boodschappen die hij kiest. De tegenstander wint het spel als hij erin slaagt een boodschap M en een handtekening σ uit te voeren zodat $\mathsf{Vf}(pk, M, \sigma) = 1$ en zodat M niet voorheen getekend werd door het orakel.

Identiteitsgebaseerde handtekeningsschema's. Een *identiteitsgebaseerd handtekeningsschema (IBS-schema)* $\mathcal{IBS}$ is een tupel van vier algoritmes (MKg, $\mathsf{UKg}, \overline{\mathsf{Sign}}, \overline{\mathsf{Vf}}$). Het hoofdsleutelgeneratie-algoritme MKg en het gebruikerssleutelgeneratie-algoritme UKg zijn gedefinieerd zoals voor IBI-schema's, het handtekeningsalgoritme $\overline{\mathsf{Sign}}$ genereert op invoer usk, M een handtekening σ, en het verificatie-algoritme $\overline{\mathsf{Vf}}$ beslist op basis van de publieke hoofdsleutel mpk, een identiteit I, een boodschap M en een handtekening σ of het handtekening geldig is of niet.

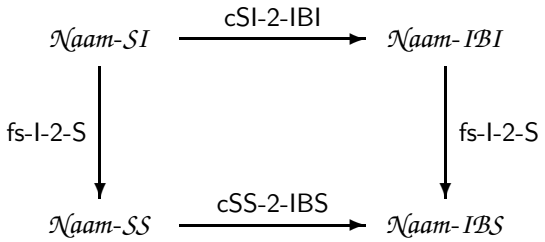
Veiligheid van IBS-schema's [CC03] is gedefinieerd op een gelijkaardige manier als SS-schema's, maar met bijkomende initialisatie- en corruptie-orakels net als in het experiment voor IBI-schema's. De IBS-tegenstander $\overline{\mathsf{F}}$ kan handtekeningen opvragen voor boodschappen en identiteiten naar keuze, en moet om het spel te winnen I, M, σ teruggeven zodat $\overline{\mathsf{Vf}}(mpk, I, M, \sigma) = 1$ en zodat M nog niet getekend was onder identiteit I door het orakel. Het voordeel $\mathbf{Adv}^{\text{uf-cma}}_{\mathcal{IBI},\overline{\mathsf{F}}}(k)$ is opnieuw de kans dat $\overline{\mathsf{F}}$ dit spel wint, en het schema is uf-cma veilig als het voordeel een verwaarloosbare functie is in k voor alle polynomiale-tijd tegenstanders $\overline{\mathsf{F}}$.

3.3 Transformaties

Als eerste stap presenteren we het raamwerk van transformaties afgebeeld in Figuur 5.1, dat (in de meeste gevallen) het bewijzen van IBI- of IBS-schema's herleidt tot het bewijzen van een onderliggend SI-schema. Sommige van deze SI-schema's waren al geanalyseerd in de literatuur, voor veel andere schema's was dit nog niet het geval. De tweede stap is veiligheidsbewijzen te voorzien voor de SI-schema's die nog niet veilig bewezen waren, en rechtstreekse veiligheidsbewijzen als IBI- of IBS-schema's te voorzien voor de zeldzame uitzonderingen die niet omvat worden door ons raamwerk.

Wij zijn van mening dat de waarde van dit raamwerk verder gaat dan het herleiden van het veilig bewijzen van IBI- en IBS-schema's tot het bewijzen van SI-schema's. Het helpt een inzicht te verwerven in het ontwerp van IBI- en IBS-schema's, en brengt tijdens dit proces de hierboven vermelde impliciete schema's aan de oppervlakte. Globaal gezien draagt het raamwerk bij tot het vereenvoudigen en verenigen van het onderzoeksdomein.

We introduceren een klasse van SI-schema's die we *converteerbaar* noemen. Het idee is dat het sleutelgeneratie-algoritme gebaseerd is op wat we een *val-*



Figuur 5.1: Familie van schema's geassocieerd met een cSI-schema $\mathcal{N}aam\text{-}\mathcal{SI}$. Als $\mathcal{N}aam\text{-}\mathcal{SI}$ imp-atk veilig is voor atk $\in \{\text{pa}, \text{aa}, \text{ca}\}$, dan is $\mathcal{N}aam\text{-}\mathcal{IBI}$ ook imp-atk veilig. Als $\mathcal{N}aam\text{-}\mathcal{SI}$ imp-pa veilig is, dan is $\mathcal{N}aam\text{-}\mathcal{IBS}$ uf-cma veilig. De structuur van het diagramma impliceert dat fs-I-2-S(cSI-2-IBI($\mathcal{N}aam\text{-}\mathcal{SI}$)) = cSS-2-IBS(fs-I-2-S($\mathcal{N}aam\text{-}\mathcal{SI}$)).

deur bemonsterbare relatie noemen. We stellen dan een transformatie cSI-2-IBI voor die een converteerbaar SI-schema (cSI-schema) omzet in een IBI-schema, en we bewijzen dat de transformatie veiligheidsbewarend is: als het oorspronkelijke cSI-schema veilig is tegen impersonatie onder passieve, actieve of concurrente aanval, dan is het resulterende IBI-schema dat ook in het willekeurig-orakelmodel.

Analoog daaraan definiëren we converteerbare SS-schema's (cSS-schema's) en een transformatie cSS-2-IBS die uf-cma veilige cSS-schema's omvormt tot uf-cma veilige IBS-schema's. Deze transformatie is een veralgemening van de transformatie van Dodis et al. [DKXY03] in de zin dat elk valdeur SS-schema tevens een cSS-schema is, en onze cSS-2-IBS transformatie identiek is aan de transformatie van Dodis et al. [DKXY03] als het oorspronkelijk schema een valdeur SS-schema is. Onze klasse van cSS-schema's is echter strikt groter dan de klasse valdeur SS-schema's.

De gekende Fiat-Shamir transformatie [FS86] zet SI-schema's om tot SS schema's, en het is geweten dat als het eerste veilig is onder passieve aanval (imp-pa), het laatste onvervalsbaar is onder gekozen-boodschap aanval (uf-cma) in het willekeurig-orakelmodel [AABN02]. Wij verwijzen naar deze transformatie als de fs-I-2-S transformatie. Toepassing van deze transformatie vereist dat het SI-schema aan bepaalde technische voorwaarden voldoet, maar deze zullen altijd voldaan zijn voor de concrete schema's die wij behandelen.

Door samenstelling van de bovenstaande transformaties kunnen we inzien dat voor elk imp-pa veilig cSI-schema $\mathcal{SI}$, het IBS-schema $\mathcal{IBS}$ = cSS-2-IBS(fs-I-2-S($\mathcal{SI}$)) uf-cma veilig is in het willekeurig-orakelmodel. We merken op dat fs-I-2-S tevens een IBI-schema omzet tot een IBS-schema, en dat het diagramma in Figuur 5.1 "commuteert", waarmee we bedoelen dat cSS-2-IBS(fs-I-2-S($\mathcal{SI}$)) = fs-I-2-S(cSI-2-IBI($\mathcal{SI}$)) voor elk cSI-schema $\mathcal{SI}$.

Terloops merken we op dat het analoog resultaat van Abdalla et al. [AABN02] in het algemeen *niet* opgaat voor fs-I-2-S als een transformatie van IBI-schema's

naar IBS-schema's: we tonen aan dat er imp-pa veilige IBI-schema's bestaan die na transformatie door fs-I-2-S onveilige IBS-schema's opleveren. Dit is niet in tegenspraak met het voorgaande, gezien het IBI-schema in kwestie niet het resultaat is van het toepassen van de cSI-2-IBI transformatie op een cSI-schema, maar compliceert de situatie lichtjes in enkele uitzonderlijke gevallen waar we een IBS-schema veilig willen afleiden van een IBI-schema dat niet de cSI-2-IBI transformatie van een cSI-schema is. Hiertoe breiden we de fs-I-2-S transformatie uit tot de efs-IBI-2-IBS transformatie die wel *elk* imp-pa veilig IBI-schema omzet in een uf-cma veilig IBS-schema.

3.4 Veiligheid van Specifieke Schema's

Families van schema's. We trachten alle IBI-schema's $\mathcal{IBI}$ in de literatuur te analyseren door een cSI-schema $\mathcal{SI}$ aan de oppervlakte te brengen zodat cSI-2-IBI($\mathcal{SI}$) = $\mathcal{IBI}$, en op een gelijkaardige manier trachten we voor alle IBS-schema's $\mathcal{IBS}$ in de literatuur een cSI-schema $\mathcal{SI}$ aan de oppervlakte te brengen zodat cSS-2-IBS(fs-I-2-S($\mathcal{SI}$)) = $\mathcal{IBS}$. We slagen hierin voor de meeste schema's die we vonden in de literatuur [FS86, GQ89, Sha84, Gir90, Hes03, CC03, Yi03, Bet88] en het RSA-gebaseerd IBI-schema van Okamoto [Oka93]. Voor deze schema's herleiden we dus de taak $\mathcal{IBI}$ en $\mathcal{IBS}$ veilig te bewijzen tot het veilig bewijzen van het $\mathcal{SI}$ schema.

We merken op dat deze manier van werken verscheidene schema's aan de oppervlakte brengt die "nieuw" zijn in de betekenis dat ze nooit expliciet in de literatuur voorgesteld zijn. Zo presenteerde Shamir [Sha84] bijvoorbeeld het IBS-schema $\mathcal{Sh}$-$\mathcal{IBS}$, maar geen IBI-schema. (Hij vermeldt zelfs het ontwerp van een IBI-schema als open probleem.) Uit zijn IBS-schema brengen wij een cSI-schema $\mathcal{Sh}$-$\mathcal{SI}$ aan de oppervlakte zodat cSS-2-IBS(fs-I-2-S($\mathcal{Sh}$-$\mathcal{SI}$)) = fs-I-2-S(cSI-2-IBI($\mathcal{Sh}$-$\mathcal{SI}$)) = $\mathcal{Sh}$-$\mathcal{IBS}$. Bijgevolg leiden we hieruit ook het IBI-schema $\mathcal{Sh}$-$\mathcal{IBI}$ = cSI-2-IBI($\mathcal{Sh}$-$\mathcal{SI}$) af, dat op natuurlijke verwant is met het $\mathcal{Sh}$-$\mathcal{IBS}$ schema, namelijk door het feit dat fs-I-2-S($\mathcal{Sh}$-$\mathcal{IBI}$) = $\mathcal{Sh}$-$\mathcal{IBS}$. Op een analoge manier brengen we ook de IBI-schema's $\mathcal{Hs}$-$\mathcal{IBI}$ en $\mathcal{ChCh}$-$\mathcal{IBI}$ aan de oppervlakte, die aan de basis liggen van de voorgestelde paring-gebaseerde IBS-schema's [Hes03, CC03, Yi03].

Naast het analyseren van bestaande IBI- en IBS-schema's, leiden we ook enkele nieuwe schema's af. We hebben publicaties in de literatuur gevonden [OO90, OS90, FF02] die geen IBI- of IBS-schema's definiëren, maar die wel SI-schema's definiëren waarvan wij aantonen dat ze converteerbaar zijn. Via onze transformaties levert dit nieuwe IBI- en IBS-schema's op waarvan we de veiligheid analyseren.

Resultaten voor specifieke schema's. Om de de nieuwe en bestaande IBI- en IBS-schema's veilig te bewijzen, rest ons enkel nog de veiligheid van onderliggende cSI-schema's te analyseren. Dit bleek echter een omvangrijke taak, want alhoewel in sommige gevallen het cSI-schema reeds bewezen werd in de litera-

Naam	Oorsprong	*Naam-SI*			*Naam-IBI*			*Naam-SS*	*Naam-IBS*
		imp-pa	imp-aa	imp-ca	imp-pa	imp-aa	imp-ca	uf-cma	uf-cma
$\mathcal{FS}$	IBI,IBS [FS86, FFS88]	[FS86]	[FFS88]	**I**	**I**	**I**	**I**	[PS00]	[DKXY03]
$\mathcal{ItR}$	SI, SS [OO90, OS90]	[Sch96]	[Sch96]	**O**	**I**	**I**	**O**	[PS00]	[DKXY03]
$\mathcal{FF}$	SI,SS [FF02]	[FF02]	[FF02]	[FF02]	**I**	**I**	**I**	[FF02]	[DKXY03]
$\mathcal{GQ}$	IBI, IBS [GQ89]	[GQ89]	[BP02]	[BP02]	**I**	**I**	**I**	[PS00]	[DKXY03]
$\mathcal{Sh}$	IBS [Sha84]	**B**	**A**	**A**	**I**	**A**	**A**	**I**	**I**
$\mathcal{Sh}^*$	SI	**B**	**B**	**B**	**I**	**I**	**I**	**I**	**I**
$\mathcal{OKRSA}$	SI, IBI, SS [Oka93]	[Oka93]	[Oka93]	**I**	**I**	**I**	**I**	[PS00]	[DKXY03]
$\mathcal{Gir}$	SI, IBI [Gir90, SSN98]	**A**	**A**	**A**	**A**	**A**	**A**	**A**	**A**
$\mathcal{SOK}$	IBS [SOK00]	**B**	**A**	**A**	**I**	**A**	**A**	**I**	**I**
$\mathcal{Hs}$	IBS [Hes03]	**B**	**B**	**B**	**I**	**I**	**I**	[Hes03]	[DKXY03]
$\mathcal{ChCh}$	IBS [CC03, Yi03]	**B**	**B**	**B**	**I**	**I**	**I**	[CC03]	[CC03]
$\mathcal{Beth}^1$	IBI [Bet88]	**B**	**O**	**O**	**I**	**O**	**O**	**I**	**I**
$\mathcal{Beth}^t$	IBI [Bet88]	**O**	**O**	**O**	**O**	**O**	**O**	**O**	**O**
$\mathcal{OKDL}$	IBI [Oka93]	**I**	**I**	**I**	**B**	**B**	**B**	**I**	**I**
$\mathcal{XDL}$	SI, IBI	**I**	**I**	**I**	**B**	**B**	**B**	**I**	**I**

Figuur 5.2: Samenvatting van veiligheidsresultaten. Kolom 1 is de familienaam van een schemafamilie. Kolom 2 vermeldt welk van de vier schema's reeds voorkwam in de literatuur. (De andere schema's brengen wij aan de oppervlakte.) In de veiligheidskolommen wordt een bekend resultaat aangeduid met een referentie naar de publicatie waarin het behaald wordt. De symbolen **I**, **B**, en **A** duiden allemaal op nieuwe resultaten behaald in deze thesis. Een **I** duidt op een veiligheidsbewijs behaald door implicatie, hetzij door toepassing van een transformatie, hetzij door eenvoudige uitbreiding van bestaand werk. Een **B** duidt op een volledig nieuw bewijs. Een **A** betekent dat we een aanval op het bewuste schema hebben gevonden, en een **O** betekent dat de veiligheid onbekend is. In alle rijen behalve de twee laatste is het SI-schema converteerbaar. De eerste groep schema's zijn gebaseerd op factorisatie, de tweede op RSA, de derde op paringsfuncties en de laatste op discrete logaritmen.

tuur, was dit vaak ook niet het geval. Bovendien zagen we ons genoodzaakt de veiligheid van twee IBI-schema's rechtstreeks te bewijzen, gezien deze niet gebaseerd waren op een onderliggend cSI-schema.

We geven een samenvatting van onze resultaten in Figuur 5.2. Merk op dat alle veiligheidsbewijzen voor SS-, IBI- en IBS-schema's in het willekeurig-orakelmodel plaatsvinden. We vatten onze resultaten hier kort samen.

Eenvoudige gevallen zijn de $\mathcal{FS}$, $\mathit{It}\mathcal{R}$, $\mathcal{FF}$, $\mathcal{GQ}$, en $\mathcal{OKRSA}$ families, waar de SI-schema's reeds voorgesteld en bewezen werden in voorgaand werk [FFS88, Sch96, FF02, BP02, Oka93].

Het $\mathcal{Sh}$-$\mathcal{SI}$ schema blijkt een "spiegelbeeld" te zijn van $\mathcal{GQ}$-$\mathcal{SI}$, en is technisch interessant omdat we aantonen dat het zero-knowledge eigenschappen bezit die het op het eerste zicht niet lijkt te hebben. Op basis van die eigenschappen bewijzen we dat het schema imp-pa veilig is onder de RSA-veronderstelling, maar een eenvoudige aanval toont aan dat het onveilig is onder actieve en concurrente aanval. Een lichte variant $\mathcal{Sh}^*$-$\mathcal{SI}$ op dit schema is echter niet enkel imp-pa onder dezelfde veronderstelling, maar is bovendien imp-aa en imp-ca veilig onder de één-meer RSA-veronderstelling, volledig gelijkaardig aan het $\mathcal{GQ}$-$\mathcal{SI}$ schema [BP02].

Het IBI-schema van Girault [Gir90] werd eerder al aangevallen en gerepareerd [SSN98], maar wij hebben nieuwe aanvallen gevonden op het gerepareerde schema en breken daarmee alle schema's in de familie.

We bewijzen imp-pa veiligheid van de paring-gebaseerde $\mathcal{SOK}$-$\mathcal{SI}$, $\mathcal{Hs}$-$\mathcal{SI}$ en $\mathcal{ChCh}$-$\mathcal{SI}$ schema's onder de CDH-veronderstelling, en imp-aa en imp-ca veiligheid onder de één-meer CDH-veronderstelling. We merken op dat het $\mathcal{SOK}$-$\mathcal{IBS}$ schema gedefinieerd via onze transformaties nauw verwant maar niet identiek is aan het gepubliceerde schema [SOK00]. Dit duidt op de waarde van ons raamwerk, gezien het onduidelijk is of het gepubliceerde IBS-schema [SOK00] uf-cma veilig kan bewezen worden, terwijl onze transformaties wel uf-cma veiligheid garanderen.

Gezien de afwezigheid van een valdeur voor discrete logaritmen, is het geen evidente keuze om IBI-schema's op te baseren. Toch bestaan er enkele: het (onbewezen) $\mathcal{Beth}^t$-$\mathcal{IBI}$ schema [Bet88] is geparametriseerd met een "sleutelmeervoudigheid" $t \in \{1, 2, \ldots\}$ en is gebaseerd op ElGamal handtekeningen [El 84]. Via een truuk slagen we erin aan te tonen dat het $\mathcal{Beth}^1$-$\mathcal{SI}$ schema dat we aan de oppervlakte brengen converteerbaar is, en we bewijzen dat $\mathcal{Beth}^1$-$\mathcal{SI}$ imp-pa veilig is onder een (vrij lichte, doch onbewezen) veronderstelling betreffende ElGamal handtekeningen. We waren niet in staat $\mathcal{Beth}^1$-$\mathcal{SI}$ te bewijzen of te breken onder actieve en concurrente aanvallen. Ook hebben we geen veiligheidsresultaten kunnen aantonen voor de $\mathcal{Beth}^t$ familie voor $t > 1$.

Uitzonderingen. De laatste twee rijen van Figure 5.2 bevatten schema's waar ons raamwerk niet van toepassing is en directe analyses noodzakelijk zijn. Het eerste is een voorheen onbewezen IBI-schema gebaseerd op discrete logaritmen

OkDL-IBI [Oka93]. Alhoewel dit schema op een natuurlijke manier kan herleid worden tot een SI-schema, is dit laatste niet converteerbaar. Desalniettemin tonen we aan dat *OkDL-IBI* veilig is onder passieve, actieve en concurrente aanval onder de discrete-logaritmeveronderstelling. Dit rechtstreekse bewijs is ongetwijfeld het meest technische in dit hoofdstuk, en illustreert nogmaals het nut van ons raamwerk dat voor de meeste gevallen de omslachtigheden van een rechtstreeks bewijs vermijdt. Bovendien presenteren we een nieuw IBI-schema *XDL-IBI* dat een iets efficiëntere variant is van het *OkDL-IBI* schema, en dat gelijkaardige veiligheidseigenchappen heeft als dit laatste.

Gezien ze niet afkomstig zijn van cSI-schema's, kunnen de IBS-schema's fs-I-2-S(*OkDL-IBS*) en fs-I-2-S(*XDL-IBS*) niet veilig bewezen worden op basis van de veiligheidseigenschappen van de IBI-schema's. We kunnen echter wel onze gewijzigde efs-IBI-2-IBS transformatie toepassen op de IBI-schema's, en verkrijgen op die manier toch uf-cma veilige IBS-schema's.

4 Transitieve Handtekeningen

4.1 Achtergrond

Het concept van transitieve handtekeningen werd geïntroduceerd door Micali en Rivest [MR02b]. Het probleem is het volgende: een ondertekenaar wil dynamisch een grafe authenticeren, verbinding per verbinding, zodat eender wie (dus niet enkel de ondertekenaar) vanuit twee handtekeningen σ_1, σ_2 voor aanliggende verbindingen $\{i, j\}$ en $\{j, k\}$ zelf een derde handtekening σ_3 kan berekenen voor de rechtstreekse verbinding $\{i, k\}$. De geauthenticeerde grafe bestaat dus niet enkel uit de verbindingen die expliciet getekend werden door de ondertekenaar, maar is de volledige transitieve sluiting hiervan.

Micali en Rivest vermelden als mogelijke toepassingen militaire hiërarchische bevelstructuren, waar elke knoop een militair personeelslid voorstelt, en een gerichte verbinding van i naar j betekent dat i het bevel voert over j; en administratieve domeinen, waar knopen machines voorstellen, en een ongerichte verbinding tussen i en j betekent dat i en j tot hetzelfde domein behoren. Een echt overtuigende toepassing moet echter nog gevonden worden, en alhoewel het waarschijnlijker is dat een dergelijke toepassingen gevonden wordt voor gerichte grafen, blijken (niet-triviale) gerichte transitieve handtekeningsschema's veel moeilijker te construeren dan ongerichte: tot nog toe werden geen ongerichte schema's voorgesteld, en in navolging van ons werk argumenteerde Hohenberger [Hoh03] zelfs dat er ofwel een nieuwe, voorlopig ongekende algebraïsche structuur nodig is om dit te bereiken, ofwel een volledig nieuwe aanpak van het probleem. Ons werk legt zich toe op ongerichte transitieve handtekeningen.

Een transitief handtekeningsschema (TS-schema) kan triviaal gerealiseerd worden door als handtekening voor een verbinding $\{i, j\}$ elke sequentie van

getekende verbindingen te aanvaarden die een pad vormen van i naar j. De onbeperkte lengte van een handtekening en het verlies aan privacy doordat de handtekeningen informatie bevatten over hun geschiedenis, sluiten deze aanpak echter uit. Het belangrijkste resultaat van Micali and Rivest [MR02b] is een (niet-triviaal) transitief handtekeningsschema, hier $\mathcal{DL}$-$\mathcal{TS}$ genoemd, dat bewijsbaar veilig is onder adaptieve aanval (zie Sectie 4.2 voor definities) onder de veronderstelling dat het discrete-logaritmeprobleem niet efficiënt oplosbaar is en dat een onderliggend SS-schema uf-cma veilig is. Ze beschrijven tevens een RSA-gebaseerd schema, hier $\mathcal{RSA}$-$\mathcal{TS}$ genoemd, en vermelden dat alhoewel het veilig lijkt en alhoewel het bewijsbaar veilig is onder *niet-adaptieve* aanval, er geen veiligheidsbewijs onder *adaptieve* aanval gekend is.

Voorafgaand aan ons werk hadden transitieve handtekeningen (veilig onder adaptieve aanval) dus slechts één enkele realisatie, namelijk het $\mathcal{DL}$-$\mathcal{TS}$ schema. Het is de gewoonte in cryptografie op zoek te gaan naar nieuwe en alternatieve realisaties van bestaande primitieven, zowel om efficiëntie-voordelen te bereiken, als om het bestaan van het primitief sterker theoretisch te onderbouwen door middel van constructies gebaseerd op alternatieve moeilijke problemen. Dit werk presenteert een aantal nieuwe schema's die beide doelen bereiken, en beantwoordt bovendien de open vraag betreffende het $\mathcal{RSA}$-$\mathcal{TS}$ schema.

4.2 Definities en Veiligheidsnoties.

Transitieve handtekeningsschema's. Alle grafen die we hier behandelen zijn ongericht. Als $G = (V, E)$ een grafe is, dan is de *transitieve sluiting* van G de grafe $\widetilde{G} = (V, \widetilde{E})$ waarbij $\{i, j\} \in \widetilde{E}$ als en slechts als er een pad is van i naar j in G. De grafe G is *transitief gesloten* als hij gelijk is aan zijn transitieve sluiting, oftewel als voor alle knopen $i, j, k \in V$ zodat $\{i, j\} \in E$ en $\{j, k\} \in E$, er ook geldt dat $\{i, k\} \in E$. Merk op dat een transitief gesloten ongerichte grafe gepartitioneerd is in disjuncte componenten zodat elke component een volledige grafe is.

Een *transitief handtekeningsschema (TS-schema)* $\mathcal{TS}$ is een tupel van vier polynomiale-tijd algoritmes $(\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ zodat:

- het gerandomiseerde sleutelgeneratie-algoritme TKg op invoer 1^k, met $k \in \mathbb{N}$ de veiligheidsparameter, een sleutelpaar teruggeeft bestaande uit de publieke sleutel *tpk* en de private sleutel *tsk*.

- het tekenalgoritme TSign als invoer de private sleutel *tsk* en twee knopen $i, j \in \mathbb{N}$ neemt, en een *originele handtekening* σ teruggeeft voor de verbinding $\{i, j\}$.

- het verificatie-algoritme TVf beslist, gegeven *tpk*, knopen $i, j \in \mathbb{N}$ en een kandidaat handtekening σ, of σ een geldige handtekening is of niet.

Schema	**Kost ondertekenen**	**Kost verifiëren**	**Kost samenstellen**	**Lengte handtekening**
$\mathcal{DL}$-$\mathcal{TS}$	2 stand. handt. 2 exp. in $\mathbb{G}$	2 stand. verifs 1 exp. in $\mathbb{G}$	2 opt. in $\mathbb{Z}_q$	2 stand. handt. 2 punten in $\mathbb{G}$ 2 punten in $\mathbb{Z}_q$
$\mathcal{DL}1\mathcal{m}$-$\mathcal{TS}$	2 stand. handt 1 exp. in $\mathbb{G}$	2 stand. verifs 1 exp. in $\mathbb{G}$	1 opt. in $\mathbb{Z}_q$	2 stand. handt. 2 punten in $\mathbb{G}$ 1 point in $\mathbb{Z}_q$
$\mathcal{RSA}$-$\mathcal{TS}$	2 stand. handt. 2 RSA encs	2 stand. verifs 1 RSA enc.	$O(\|N\|^2)$ ops	2 stand. handt. 3 punten in $\mathbb{Z}_N^*$
$\mathcal{F}act$-$\mathcal{TS}$	2 stand. handt. $O(\|N\|^2)$ ops	2 stand. verifs $O(\|N\|^2)$ ops	$O(\|N\|^2)$ ops	2 stand. handt. 3 punten in $\mathbb{Z}_N^*$
$\mathcal{G}ap$-$\mathcal{TS}$	2 stand. handt. 2 exp. in $\hat{\mathbb{G}}$	2 stand. verifs 1 $\mathsf{S}_{\mathrm{ddh}}$	$O(\|N\|^2)$ ops	2 stand. handt. 3 punten in $\hat{\mathbb{G}}$
$\mathcal{RSAH}$-$\mathcal{TS}$	1 RSA dec.	1 RSA enc.	$O(\|N\|^2)$ ops	1 punt in $\mathbb{Z}_N^*$
$\mathcal{F}act\mathcal{H}$-$\mathcal{TS}$	2 vkwortels in $\mathbb{Z}_N^*$	$O(\|N\|^2)$ ops	$O(\|N\|^2)$ ops	1 punt in $\mathbb{Z}_N^*$
$\mathcal{G}ap\mathcal{H}$-$\mathcal{TS}$	1 exp. in $\hat{\mathbb{G}}$	1 $\mathsf{S}_{\mathrm{ddh}}$	$O(\|N\|^2)$ ops	1 punt in $\hat{\mathbb{G}}$

Figuur 5.3: Kostenvergelijking tussen transitieve handtekeningsschema's. Het woord "stand." verwijst naar operaties van het onderliggende standaard handtekeningsschema, dat niet van toepassing is voor $\mathcal{RSAH}$-$\mathcal{TS}$, $\mathcal{F}act\mathcal{H}$-$\mathcal{TS}$ en $\mathcal{G}ap\mathcal{H}$-$\mathcal{TS}$. $\mathbb{G}$ stelt de discrete-logaritmegroep voor van $\mathcal{DL}$-$\mathcal{TS}$ en $\mathcal{DL}1\mathcal{m}$-$\mathcal{TS}$, en N is een product van twee priemgetallen zoals gebruikt in RSA en factorisatie-gebaseerde schema's. $\hat{\mathbb{G}}$ is een kloof-Diffie-Hellmangroep en $\mathsf{S}_{\mathrm{ddh}}$ is een uitvoering van het DDH-algoritme in $\hat{\mathbb{G}}$. Gebruikte afkortingen zijn: "exp." voor een machtsverheffing in de groep; "RSA enc." voor een RSA encryptie; "RSA dec." voor een RSA decryptie met gegeven decryptie-exponent; "vkwortel" voor een vierkantswortel modulo N gegeven de priemfactoren van N; en "ops" voor het aantal elementaire bit-operaties in grote-O notatie.

- het samenstellingsalgoritme Comp de publieke sleutel tpk, knopen $i, j, k \in \mathbb{N}$ en handtekeningen σ_1, σ_2 voor verbindingen $\{i, j\}$ en $\{j, k\}$ als invoer neemt, en een handtekening σ_3 voor verbinding $\{i, k\}$ teruggeeft (of een symbool $\perp$ om aan te duiden dat het samenstellen gefaald is).

Correctheid van TS-schema's. Natuurlijk verwachten we dat als σ een originele handtekening is ten opzichte van tsk, dat dan σ ook een geldige handtekening is ten opzichte van tpk. Een sluitende definitie vinden voor de correctheid van het samenstellingsalgoritme is echter minder evident. Volgens de definitie van Micali en Rivest [MR02b] mag de samenstelling van twee originele handtekeningen niet te onderscheiden zijn van een originele handtekening voor dezelfde verbinding. Dit lijkt echter te impliceren dat samenstelling enkel werkt voor originele handtekeningen, en recursieve samenstelling dus onmogelijk is.

We zouden, in navolging van Johnson et al. [JMSW02], kunnen eisen dat de samenstelling van twee geldige handtekeningen steeds een derde geldige handtekening oplevert, maar deze vereiste blijkt te sterk voor onze doeleinden: zowel voor het $\mathcal{DL}$-$\mathcal{TS}$ schema als voor onze schema's bestaan er geldige handtekeningen die na samenstelling een ongeldig handtekening opleveren. Het vinden van zulke handtekeningen vereist wel het breken van het onderliggende SS-schema en is dus computationeel onhaalbaar, maar toch zouden we de correctheids- en veiligheidsdefinities liever niet met mekaar verweven. Daarom formuleren we een correctheidsdefinitie via een recursieve vereiste die zegt dat zolang de originele handtekeningen "legitiem" verkregen zijn, d.w.z. rechtstreeks van de ondertekenaar of door samenstelling van legitieme handtekeningen, de resulterende handtekening geldig moet zijn.

Veiligheid van TS-schema's. Een vervalser voor een TS-schema $\mathcal{TS} = (\mathsf{TKg}, \mathsf{TSign}, \mathsf{TVf}, \mathsf{Comp})$ is een algoritme F dat de publieke sleutel tpk als invoer krijgt en een toegang heeft tot een handtekeningsorakel $\textsc{TSign}(\cdot,\cdot) = \mathsf{TSign}(tsk, \cdot, \cdot)$, waaraan F handtekeningen kan opvragen voor verbindingen naar zijn keuze. Laat E de verzameling van randen $\{i, j\}$ zijn waarvoor F een handtekening heeft gevraagd, en laat V de verzameling zijn van alle betrokken knopen. Uiteindelijk geeft F knopen $i', j' \in \mathbb{N}$ en een vervalst handtekening σ' terug. De vervalser wint het spel als $\mathsf{TVf}(tpk, i', j', \sigma') = 1$ terwijl $\{i', j'\}$ niet binnen de transitieve sluiting van $G = (V, E)$ ligt. Het schema $\mathcal{TS}$ is transitief onvervalsbaar onder adaptieve gekozen-boodschap aanval (tu-cma veilig) als geen enkele polynomiale-tijd tegenstander F een niet-verwaarloosbare kans heeft om dit spel te winnen. Een zwakkere veiligheidsnotie voor TS-schema's is transitieve onvervalsbaarheid onder *niet-adaptieve* gekozen-boodschap aanval. Hierbij moet de vervalser op voorhand beslissen welke verbindingen hij wil laten tekenen, en kan hij zijn volgende bevraging dus niet laten afhangen van de waarde van de vorige handtekeningen.

De knoopcertificatie-techniek. De aanpak van zowel het $\mathcal{DL}$-$\mathcal{TS}$ als het

$\mathcal{RSA}$-$\mathcal{TS}$ schema is gebaseerd op knoopcertificaten. De private sleutel van de ondertekenaar bevat de private sleutel van een SS-schema, en de publieke sleutel bevat onder meer de overeenkomstige publieke sleutel van het SS-schema. De ondertekenaar associeert met elke knoop i in de grafe een *knoopcertificaat* bestaande uit een *publiek label* $L(i)$ en een standaard handtekening op $i \| L(i)$. Het handtekening voor een verbinding bevat de knoopcertificaten van de eindpunten en *verbindingslabel* δ. Verificatie van een handtekening gebeurt door de standaard handtekeningen in de knoopcertificaten te verifiëren en de waarde van het verbindingslabel te relateren aan die van de publieke labels. Samenstelling van handtekeningen gebeurt door algebraïsche bewerkingen op de verbindingslabels.

Deze techniek is nuttig, maar brengt kosten met zich mee. Het tekenen van een verbinding beslaat het berekenen van twee standaard handtekeningen, en de lengte van een verbindingshandtekening, dat de twee knoopcertificaten bevat, kan groot worden, ook al zijn de verbindingslabels vrij klein.

4.3 Transitieve Handtekeningen op basis van RSA

Het $\mathcal{RSA}$-$\mathcal{TS}$ schema werd kort vermeld [MR02b] en volgt de knoopcertificatietechniek. Men kan bewijzen dat $\mathcal{RSA}$-$\mathcal{TS}$ transitief onvervalsbaar is onder *niet-adaptieve* gekozen-boodschap aanval onder de veronderstelling dat het RSA-probleem moeilijk is en het onderliggende SS-schema veilig is. Er werd geen adaptieve aanval op het schema gevonden, maar evenmin werd een veiligheidsbewijs onder adaptieve aanval gegeven.

Deze situatie (namelijk een schema dat zowel aanval als bewijs lijkt te weerstaan) is niet ongewoon in de cryprografie, en wij opperen dat ze te wijten is aan het feit dat de veiligheid van het schema gebaseerd is op eigenschappen van RSA die verder gaan dan gewone éénwegsheid. In het licht hiervan zijn we op zoek gegaan naar zwaardere veronderstellingen voor de RSA-functie, en hebben we kunnen aantonen dat $\mathcal{RSA}$-$\mathcal{TS}$ transitief onvervalsbaar is onder *adaptieve* gekozen-boodschap aanval als het één-meer RSA-probleem moeilijk is en het onderliggende SS-schema veilig is.

4.4 Nieuwe Transitieve Handtekeningsschema's

Het $\mathcal{Fact}$-$\mathcal{TS}$ schema. Na het $\mathcal{RSA}$-$\mathcal{TS}$ schema gezien te hebben, zou men zich kunnen afvragen of er een TS-schema bestaat dat bewijsbaar tu-cma veilig is onder de veronderstelling dat het gewone RSA-probleem moeilijk is. We beantwoorden deze vraag in positieve zin door het $\mathcal{Fact}$-$\mathcal{TS}$ schema voor te stellen en veilig te bewijzen onder de (nog zwakkere) factorisatieveronderstelling. Het bewijs bevat een delicaat informatie-theoretisch lemma dat garandeert dat vorige handtekeningen de aanvaller geen informatie geven over welke van twee

Schema	Bewijsbaar veilig onder *adaptieve* gekozen-boodschap aanval onder veronderstelling dat	WO?
$\mathcal{DL}$-$\mathcal{TS}$	Veiligheid van SS-schema Moeilijkheid van discrete logaritmen	Nee
$\mathcal{DL}1m$-$\mathcal{TS}$	Veiligheid van SS-schema Moeilijkheid van één-meer discrete logaritmen	Nee
$\mathcal{RSA}$-$\mathcal{TS}$	Veiligheid van SS-schema Moeilijkheid van één-meer RSA-probleem	Nee
$\mathcal{F}act$-$\mathcal{TS}$	Veiligheid van SS-schema Moeilijkheid van factoriseren	Nee
$\mathcal{G}ap$-$\mathcal{TS}$	Veiligheid van SS-schema Moeilijkheid van één-meer CDH-probleem in kloof-DH-groepen	Nee
$\mathcal{RSAH}$-$\mathcal{TS}$	Moeilijkheid van één-meer RSA-probleem	Ja
$\mathcal{F}act\mathcal{H}$-$\mathcal{TS}$	Moeilijkheid van factoriseren	Ja
$\mathcal{G}ap\mathcal{H}$-$\mathcal{TS}$	Moeilijkheid van één-meer CDH-probleem in kloof-DH-groepen	Ja

Figuur 5.4: Bewijsbare veiligheidseigenschappen van transitieve handtekeningsschema's. We vermelden de veronderstellingen onder dewelke een veiligheidsbewijs voor transitieve onvervalsbaarheid onder *adaptieve* gekozen-boodschap aanval bestaat, en of het willekeurig-orakelmodel gebruikt wordt in het bewijs.

vierkantswortels de ondertekenaar in gedachten heeft.

Met betrekking tot kosten geassocieerd aan een TS-schema zijn we geïnteresseerd in de computationele kost van het tekenen van een verbinding, van het verifiëren van een handtekening en van het samenstellen van twee handtekeningen, en in de lengte van een handtekening. Figuren 5.3 en 5.4 vatten respectievelijk de kosten en bewijsbare veiligheidseigenschappen van de verschillende schema's samen.

Daar $\mathcal{F}act$-$\mathcal{TS}$ eveneens de knoopcertificatie-techniek volgt, beloopt het dezelfde kosten als de $\mathcal{DL}$-$\mathcal{TS}$ en $\mathcal{RSA}$-$\mathcal{TS}$ schema's door het gebruik van het SS-schema. Zoals Figuur 5.3 illustreert is het echter computationeel goedkoper dan $\mathcal{DL}$-$\mathcal{TS}$ en $\mathcal{RSA}$-$\mathcal{TS}$ wat betreft het tekenen en verifiëren, omdat het de kosten terugbrengt van kubisch (machtsverheffing) tot kwadratisch (vermenigvuldigingen en een inverse).

Het $\mathcal{DL}1m$-$\mathcal{TS}$ schema. Het $\mathcal{DL}$-$\mathcal{TS}$ schema [MR02b] gebruikt twee generatoren. Wij beschrijven een iets eenvoudigere variant, $\mathcal{DL}1m$-$\mathcal{TS}$ genaamd, die

slechts één generator gebruikt. Zoals geïllustreerd in Figuren 5.3 en 5.4 biedt het enkele lichte performantieverbeteringen ten opzichte van $\mathcal{DL}$-$\mathcal{TS}$, maar is het slechts bewijsbaar veilig onder de moeilijkheid van het één-meer discrete-logaritmeprobleem.

Het $\mathcal{Gap}$-$\mathcal{TS}$ schema. We presenteren tevens een TS-schema gebaseerd op kloof-Diffie-Hellmangroepen genaamd $\mathcal{Gap}$-$\mathcal{TS}$, en bewijzen dat het veilig is onder de één-meer CDH-veronderstelling. Dit schema heeft eigenlijk geen rechtstreeks belang, omdat het moet onderdoen voor $\mathcal{DL1m}$-$\mathcal{TS}$ zowel qua veiligheidsveronderstellingen en performantie. De waarde van het $\mathcal{Gap}$-$\mathcal{TS}$ schema is dat er, in tegenstelling tot $\mathcal{DL1m}$-$\mathcal{TS}$ en $\mathcal{DL}$-$\mathcal{TS}$, een hash-gebaseerde verandering op kan toegepast worden die we hierna beschrijven, en we zullen zien dat het resulterende schema $\mathcal{Gap\mathcal{H}}$-$\mathcal{TS}$ de kortste handtekeningen zal hebben van alle schema's die we behandelen.

4.5 Eliminatie van Knoopcertificaten door Hashfuncties

Het $\mathcal{RSA\mathcal{H}}$-$\mathcal{TS}$ schema. Met behulp van hashfuncties kunnen we de knoopcertificaten en alle daarbij horende kosten elimineren uit het $\mathcal{RSA}$-$\mathcal{TS}$ schema. De techniek bestaat erin het publieke label van een knoop i niet te laten kiezen door de ondertekenaar, maar te definiëren als de uitvoer van een publieke hashfunctie toegepast op i, en RSA decryptie te gebruiken om verbindingslabels te berekenen. We bewijzen dat het resulterende schema $\mathcal{RSA\mathcal{H}}$-$\mathcal{TS}$ tu-cma veilig is onder de één-meer RSA-veronderstelling in het willekeurig-orakelmodel.

Het $\mathcal{Fact\mathcal{H}}$-$\mathcal{TS}$ schema. Het feit dat kwadrateren modulo een samengesteld getal een valdeur-éenwegsfunctie is, maakt het $\mathcal{Fact\mathcal{H}}$-$\mathcal{TS}$ schema geschikt voor een gelijkaardige eliminatie van de knoopcertificaten. We introduceren het $\mathcal{Fact\mathcal{H}}$-$\mathcal{TS}$ schema waar het publieke label $L(i)$ van een knoop i opnieuw bepaald wordt door de uitvoer van een hashfunctie geëvalueerd op i. We bewijzen dat het $\mathcal{Fact\mathcal{H}}$-$\mathcal{TS}$ schema tu-cma veilig is in het willekeurig-orakelmodel onder de veronderstelling dat de onderliggende modulus moeilijk te factoriseren is.

Zoals vermeld in Figuur 5.3 is de belangrijkste kostenbesparing het vermijden van alle kosten gerelateerd aan het SS-schema. Voor het ondertekenen moeten nu echter vierkantswortels berekend worden, hetgeen qua kost overeenkomt met een machtsverheffing modulo N.

Het $\mathcal{Gap\mathcal{H}}$-$\mathcal{TS}$ schema. Ook het $\mathcal{Gap}$-$\mathcal{TS}$ schema kan bevrijd worden van de knoopcertificaten, hetgeen het $\mathcal{Gap\mathcal{H}}$-$\mathcal{TS}$ schema oplevert waarvan de eigenschappen getoond worden in Figuur 5.3. De handtekening bestaat uit een enkel groepselement, en door de compacte representatie van kloof-DH-groepen betekent dit dat het $\mathcal{Gap\mathcal{H}}$-$\mathcal{TS}$ schema de kortste handtekeningen heeft van alle besproken schema's.

5 Conclusie

In deze thesis hebben we nieuwe bewijsbaar veilige schema's voorgesteld voor een aantal cryptografische problemen, en hebben we veiligheidsbewijzen opgesteld voor bestaande schema's waarvoor een dergelijk bewijs ontbrak. Waar mogelijk hebben we algemeen aanvaarde veiligheidsnoties gehanteerd, en in de enkele gevallen dat dergelijke noties niet voorhanden waren hebben we zelf nieuwe, nuttige maar haalbare veiligheidsnoties geformuleerd. We hebben ook abstracte constructies en transformaties voorgesteld die het bewijzen van concrete schema's aanzienlijk vergemakkelijken, en die bovendien bijdragen aan het begrip van de algemene principes die aan de basis ligt van het ontwerp van gerelateerde schema's.

Meer specifiek hebben we in Sectie 3 onze resultaten voor identiteitsgebaseerde identificatieschema's en handtekeningsschema's samengevat, en die voor transitieve handtekeningsschema's in Sectie 4. In wat volgt geven we nog enkele suggesties voor verder onderzoek.

Opvullen van leemtes in Figuur 5.2. Ondanks de aanzienlijke inspanningen die we gespendeerd hebben aan het onderzoeken van de veiligheidseigenschappen van schema's in Sectie 4, blijven een aantal vakjes in Figuur 5.2 onbeantwoord. Om de veiligheid van het $\mathit{It}\mathcal{R}\text{-}\mathcal{SI}$ schema onder concurrente aanval te onderzoeken zou men de details van het bewijs onder parallelle aanval [Sch96] moeten uitspitten en verifiëren of een gelijkaardig argument opgaat voor concurrente aanvallen.

Een tweede reeks open vakjes betreft de $\mathcal{B}\mathit{eth}^t$ familie. Voor het bijzondere geval van het $\mathcal{B}\mathit{eth}^1\text{-}\mathcal{SI}$ schema zijn we erin geslaagd veiligheid onder passieve aanval (en daardoor veiligheid van het bijhorend SS- en IBS-schema) te bewijzen, maar de veiligheid onder actieve en concurrente aanval blijft een open probleem. Voor de $\mathcal{B}\mathit{eth}^t$ familie met $t > 1$ is het niet eens zeker of het SI-schema converteerbaar is, en is zelfs veiligheid onder passieve aanval een open probleem.

Strakkere reducties door rechtstreekse bewijzen. Het raamwerk van transformaties in Figuur 5.1 is een krachtig hulpmiddel om asymptotische veiligheid van IBI- en IBS-schema's aan te tonen, maar de algemeenheid ervan verhindert schema-specifieke optimizaties om een "strakkere" reductie te bekomen. De reductie in een bewijs wordt *strak* genoemd als de slaagkans om het onderliggende primitief te breken ongeveer even groot is als die om het voorgestelde schema te breken. Dit is echter niet het geval voor de algemene bewijzen in ons raamwerk, hetgeen tot uiting komt als we concrete waarden invullen in de reductievergelijkingen. Om bijvoorbeeld het $\mathcal{Sh}\text{-}\mathit{IBS}$ schema even veilig te maken als het $\mathcal{Sh}\text{-}\mathcal{SI}$ schema met een 1024-bit sleutel tegen een tegenstander die zijn beide willekeurige orakels 2^{60} maal mag bevragen en 2^{30} handtekeningen mag opvragen, moeten we het IBS-schema in principe instantiëren met een sleutel van 6701 bits. Dit betekent *niet* dat het schema onveilig is voor kleinere

sleutels, maar betekent wel dat we er strikt gezien geen uitspraak over kunnen doen. Strakkere bewijzen kunnen wellicht bekomen worden door de techniek van Coron [Cor00] toe te passen op rechtstreekse bewijzen voor concrete IBI en IBS-schema's.

Identiteitsgebaseerde cryptografie zonder willekeurige orakels. Het feit dat de veiligheidsbewijzen van zowel de cSI-2-IBI en de cSS-2-IBS transformaties in het willekeurig-orakelmodel plaatsvinden lijkt geen toeval te zijn. Afgezien van enkele triviale maar inefficiënte oplossingen, hebben alle gekende identiteitsgebaseerde schema's een willekeurig orakel nodig om de (niet-willekeurige) identiteitsstring om te zetten naar een uniform verdeeld element van een bepaalde verzameling. Alhoewel redelijk efficiënte schema's zonder willekeurige orakels bestaan voor andere primitieven zoals standaard publieke-sleutel encryptie [CS98] en handtekeningen [CS00, GHR99], bestaan zulke identiteitsgebaseerde schema's enkel onder de vorm van triviale oplossingen (zoals bijvoorbeeld IBI- en IBS-schema's) of zelfs helemaal niet (zoals bijvoorbeeld identiteitsgebaseerde encryptie). Met het oog op de bezwaren tegen het willekeurig-orakelmodel geformuleerd in Sectie 1, zou het interessant zijn het bestaan te onderzoeken van praktische en efficiënte identiteitsgebaseerde cryptografie in het standaard model.

Gerichte transitieve handtekeningen. Alle TS-schema's die we besproken hebben in Sectie 4 werken enkel voor ongerichte grafen. Als echt overtuigende toepassingen van transitieve handtekeningen gevonden worden, zullen deze waarschijnlijker betrekking hebben op gerichte grafen dan op ongerichte. Op dit ogenblik bestaan er geen constructies voor gerichte TS-schema's, en Hohenberger [Hoh03] argumenteert zelfs dat dergelijke schema's zeer moeilijk te construeren kunnen zijn: zij toont aan dat het bestaan van een dergelijk schema een algebraïsche structuur impliceert waarvan voorlopig geen voorbeelden gekend zijn. Haar resultaat geldt echter enkel voor schema's die de knoopcertificatietechniek volgen, en het is niet ondenkbaar dat gerichte TS-schema's bestaan volgens een totaal andere aanpak, zonder daarvoor exotische wiskundige structuren te moeten impliceren.

Comprimeren van certificaatketens. Certificaatketens worden gebruikt om de geldigheid van een publieke sleutel na te gaan aan de hand van een vertrouwd *wortelcertificaat* dat typisch ingebed wordt in de software van de gebruiker. Bij hiërarchisch gestructureerde publieke-sleutelinfrastructuren (PKI) tekent elke certificatie-authoriteit (CA) de publieke sleutel van de volgende. Een certificaatketen die de publieke sleutel pk_n van een gebruiker relateert aan een wortelcertificaat met publieke sleutel pk_0 bevat n handtekeningen en n publieke sleutels, als volgt:

$$pk_n \parallel \mathsf{Sign}(sk_{n-1}, pk_n) \parallel pk_{n-1} \parallel \mathsf{Sign}(sk_{n-2}, pk_{n-1}) \parallel \ldots \parallel pk_1 \parallel \mathsf{Sign}(sk_0, pk_1).$$

Ondanks de analogie tussen grafen en CA-bomen, kunnen transitieve handtekeningen niet toegepast worden om deze ketens te comprimeren tot een enkele handtekening, omdat de samen te stellen handtekeningen getekend zijn onder verschillende sleutels.

Zogenaamde *aggregaathandtekeningen* [BGLS03, LMRS] zijn beter geschikt, maar hebben nog steeds *alle* publieke sleutels nodig tijdens de verificatie, hetgeen resulteert in de aanzienlijk kortere maar nog steeds lineaire certificaatketen

$$pk_n \parallel pk_{n-1} \parallel \ldots \parallel pk_1 \parallel \sigma \ .$$

Uiteindelijk zouden we de certificaatketen nog verder willen reduceren tot

$$pk_n \parallel \sigma,$$

waar σ kan geverifieerd worden met behulp van pk_n en pk_0 alleen. Daartoe hebben we een primitief nodig met een speciaal soort samenstellingsfunctie die toelaat een "sleutelpaar ertussenuit te knijpen": gegeven een handtekening voor boodschap M onder sk_1 en een handtekening voor pk_1 onder pk_2, moet het mogelijk zijn een derde handtekening te construeren die boodschap M rechtstreeks authenticeert onder pk_2. Er zijn echter geen constructies gekend die een dergelijke functionaliteit bieden.