



IST-2002-507932

ECRYPT

European Network of Excellence in Cryptology

Network of Excellence

Information Society Technologies

D.AZTEC.3

New Technical Trends in Asymmetric Cryptography

Due date of deliverable: 31. July 2005

Actual submission date: 15. June 2005

Start date of project: 1 February 2004

Duration: 4 years

Lead contractor: Katholieke Universiteit Leuven (KUL)

Revision 1.1

Project co-funded by the European Commission within the 6th Framework Programme		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission services)	
RE	Restricted to a group specified by the consortium (including the Commission services)	
CO	Confidential, only for members of the consortium (including the Commission services)	

New Technical Trends in Asymmetric Cryptography

Editor

Louis Goubin (Axalto)

Contributors

Michel Abdalla (ENS), Jan Camenisch (IBM),
Sébastien Canard (FT R&D), Dario Catalano (ENS),
Jean-Sébastien Coron (Gemplus), Nicolas Courtois (Axalto),
Steven Galbraith (RHUL), Clemente Galdi (UNISA), Louis Goubin (Axalto),
Louis Granboulan (ENS), Fabien Laguillaumie (Caen University),
Tanja Lange (RUB), John Malone-Lee (Bristol),
Gregory Neven (KUL), Pascal Paillier (Gemplus),
Giuseppe Persiano (UNISA), Birgit Pfitzmann (IBM),
David Pointcheval (ENS), Ivan Visconti (ENS)

15. June 2005

Revision 1.1

The work described in this report has in part been supported by the Commission of the European Communities through the IST program under contract IST-2002-507932. The information in this document is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Contents

1	Summary	1
2	Signatures with special properties	3
2.1	Short signatures	3
2.1.1	Motivation and Applications	3
2.1.2	Main Categories of Schemes and Their Particularities	7
2.2	Transitive signatures	7
2.2.1	The concept	7
2.2.2	Realizing the concept	7
2.2.3	Open problems	9
2.3	Aggregate signatures	11
2.3.1	Introduction	11
2.3.2	Notations	11
2.3.3	Aggregate Signatures	11
2.4	Fail-stop signatures	13
2.4.1	Introduction	13
2.4.2	Description	14
2.4.3	State of the Art	15
2.4.4	Open Problems	16
2.5	Key-evolving signatures	18
2.5.1	The key-exposure problem	18
2.5.2	Forward-secure signature schemes	19
2.5.3	Key-insulated signature schemes	21
2.6	Blind signatures	23
2.6.1	Motivation: Electronic Cash	23
2.6.2	Security Notions	25
2.6.3	Security Results	26
2.6.4	Fair Blind Signatures	28
2.7	Undeniable signatures	30
2.7.1	Introduction	30
2.7.2	Definition	31

2.7.3	Convertible and delegated signatures	32
2.7.4	Proposed schemes	32
2.7.5	Non Transferable signatures	32
2.7.6	Open Problems	33
2.8	Group/ring signatures	35
2.8.1	Introduction	35
2.8.2	Description of the problem	36
2.8.3	State of the art	37
2.8.4	Open Problems	38
3	Encryption with special properties	40
3.1	Searchable encryption	40
3.1.1	Description and Motivations	40
3.1.2	State of the art	42
3.1.3	Open problems	43
3.2	Plaintext aware encryption	44
3.2.1	Description and Motivations	44
3.2.2	State of the art	46
3.2.3	Open problems	47
3.3	Verifiable encryption	48
3.3.1	Introduction	48
3.3.2	Applications	49
3.3.3	Security Definition	51
3.3.4	Proposed Schemes	53
3.3.5	Open Problems	54
4	Signcryption	57
4.1	Description and motivation	57
4.1.1	Introduction	57
4.1.2	Signcryption schemes	57
4.2	State of the art	58
4.2.1	Security models	58
4.2.2	Schemes	59

- 4.3 Open problems 60
- 5 Homomorphic schemes 61**
 - 5.1 Introduction 61
 - 5.2 An Overview of Known Constructions 62
 - 5.2.1 The Early Mechanisms (80's) 62
 - 5.2.2 Improved Constructions (90's) 63
 - 5.2.3 Extensions of Paillier's Scheme 63
 - 5.3 Cryptographic Applications of Homomorphic Encryption 64
 - 5.4 Further Research: Algebraic Encryption 64
- 6 Identity-based cryptography 67**
 - 6.1 Description and Motivations 67
 - 6.1.1 Notions of security for identity based encryption 67
 - 6.1.2 Identity-based identification and signatures 68
 - 6.2 State of the art 68
 - 6.2.1 Pairing 68
 - 6.2.2 Identity Based Encryption with Random Oracle 69
 - 6.2.3 Identity Based Encryption without Random Oracle 70
 - 6.2.4 Hierarchical Identity Based Encryption 71
 - 6.2.5 Fuzzy Identity Based Encryption 72
 - 6.2.6 Identity Based Identification and Signature 72
 - 6.3 Open problems 73

1 Summary

The present report deals with asymmetric cryptosystems with “special properties”. Starting from the usual classification of fundamental cryptographic primitives (encryption, signature and key agreement), we present current trends which are technical answers to new security needs in specific scenarios.

We split these schemes into five main categories:

1. Signatures

- Short signatures: These are required in environments with space and bandwidth constraints. When a human is asked to manually key in the signature, the shortest possible signature is needed.
- Transitive signatures: In this scenario we have an algorithm to sign the edges of an (undirected) graph. If two consecutive edges (i, j) and (j, k) are signed, then a signature of the edge (i, k) can be publicly obtained from the two previous signatures. Military chains of command and administrative domains were mentioned as practical applications of transitive signatures.
- Aggregate signatures: An aggregation algorithm outputs a compressed short signature from a set of signatures produced by a group of users. This aggregation of signatures can be done by anyone. Moreover, there is an aggregate verification algorithm to decide whether the aggregate signature is valid. Aggregate signature scheme has use in the secure border gateway protocol for compressing the list of signatures on distinct messages issued by distinct parties.
- Fail-stop signatures: The problem solved with this special type of signature schemes is the case where an adversary has the computational power to break the scheme. Under the hypothesis that the adversary is not able to learn the private key, the user can prove that the adversary exists by publishing a solution for a hard problem. Therefore the name *fail-stop*. In case of a failure, the user can detect it, convince other of this failure, and stop using the public key.
- Key-evolving signatures: Their goal is to reduce the potential damage in case secrets are exposed. The main idea in this case is to ensure that secrets are used only for short time periods, and that compromise of a secret does not affect anything based on secrets from other time periods. One of the challenges in designing such a system is to be able to change secret information without the inconvenience of changing public information, such as the public key.
- Blind Signature: The blind signature protocols enable a user to obtain a signature from a signer so that the signer does not learn any information about the message it signed and so that the user can not obtain more than one valid signature after one interaction with the signer. The concept of blind signatures provides anonymity of users in applications such as electronic voting, electronic payment systems, etc.
- Undeniable signatures: Unlike handwritten signatures, digital signatures can be “copy-cloned”, and therefore, authenticated documents can be easily disseminated. However, in some applications, it is necessary to allow the signer (or a third party)

to control the verification of the signature. In the context of *unverifiable signatures*, someone who wants to verify a signature has to interact with the signer.

- **Ring Signature:** For a given set of users, a ring signature is a signature that is constructed using all the public keys of the users, and a single private key of any user. A ring signature protects the anonymity of a signer since the verifier knows that the signature is from a member of the ring, but does not know exactly who the signer is. There is also no way to revoke the anonymity of the signer. Ring signatures have applications in authenticated (yet repudiable) communication and leaking secrets.
- **Group Signature:** Group signatures permits any member of a group to sign on behalf of the group. Anyone can verify the signature with a group public key while no one can know the identity of the signer except the group manager. Group signature provides anonymity of users with the property that group manager can identify the signer. In group signature, it is computationally hard to decide whether two different signatures were issued by the same member.

2. Encryption

- **Searchable encryption:** Suppose Alice wishes to read her email on a number of devices : laptop, desktop, pager, etc. Alices mail gateway is supposed to route email to the appropriate device based on the keywords in the email. Suppose Bob sends an email with keyword “urgent”. The gateway routes the email to Alices pager, after testing whether the email contains this keyword “urgent” without learning anything else about the mail. This mechanism is referred to as *searchable encryption*.
- **Plaintext aware encryption:** Intuitively, an encryption scheme is *plaintext aware* if the only way that an adversary can produce a valid ciphertext is to apply the encryption algorithm to the public key and a message. Plaintext awareness is the strongest known form of encryption. In particular, it immediately implies security against adaptive chosen-ciphertext attack and appears to be strictly stronger.
- **Verifiable encryption:** This concept deals with the general problem of proving properties about encrypted data. In the case of public-key encryption, which is the setting in which we are interested here, there are two parties who are in a position to prove some property to another party about an encrypted message namely, the party who created the ciphertext, and the party who holds the secret key. A protocol in which the encryptor is the prover is a *verifiable encryption* scheme.

3. **Signcryption** In a signcryption scheme, encryption and signature are performed in a single logical step in order to obtain confidentiality, integrity, authentication and non-repudiation more efficiently than the sign-then-encrypt approach.

4. **Homomorphic schemes** In such a scheme, encryption preserves a specific relation, in the sense that given several ciphertexts, one may compute another ciphertext whose plaintext is related to the plaintexts corresponding to the ciphertexts. For instance, in the homomorphic Paillier scheme, given two ciphertexts of two plaintexts, one can easily compute the ciphertext of the sum of the two plaintexts. Such schemes are useful in a voting scheme.

5. **Identity-based cryptography** An identity-based cryptosystem has the property that each user's public key derivable from some known aspect of his identity (such as his email address), while his private key can be calculated for him by a trusted authority. The identity-based public key cryptosystem can be an alternative for certificate-based public key infrastructure (PKI), especially when efficient key management and moderate security are required. ID-based signature schemes have been known since 1984, but only recently have ID-based encryption schemes shown feasible, using the notion of pairing.

The report is organized as follows: for each scheme, we give:

- The motivations (including needs from industry)
- Description of the problem
- State of the art, including description of the solution, security proofs, performance aspects, references, ...
- Open problems

2 Signatures with special properties

2.1 Short signatures

When we talk about short signatures, we usually (but not always) talk about “usual” signatures with appendix with an additional property of having a short length, for example only 160 or 128 bits. The security requirements will (unless otherwise stated) do not differ from the usual Goldwasser, Micali and Rivest notion of Digital Signatures [1].

2.1.1 Motivation and Applications

The main (and maybe the only) motivation for studying short signatures are various interesting applications they may have. We will overview these applications and try to explain in details what do they require, and what do they imply in terms of cryptographic primitives.

The need of short signature arises in several low-bandwidth environments in which the size of the signature is critical. The main sources of low bandwidth are:

1. Human interaction: the user has to copy, type-in, read, and transmit the signature over the phone, and other similar situation.
2. Robust machine-readable documents: one and two dimensional bar-codes, machine-readable ID-cards, passports, bank notes and cheques, documents that are digitally signed and send by fax, etc.
3. Lack of bandwidth due to the specification: for example signing individual packets of fixed size that are transmitted over the internet, some packets may never be received.

4. Lack of bandwidth due to the constrained environments: for example a limited amount and speed of E²PROM memory contained in a smart card.

Now we will list the most interesting (but not all) applications that require short signatures.

1. Electronic airline tickets.

Electronic tickets are now delivered at a massive scale, both as a result of selling more and more tickets over the internet, and as an attempt to cut on unnecessary costs in the time of market slowdown due to the threat of terrorist attacks. These tickets bear no proof of validity to the eye of the user, and in fact they have no proof of authenticity whatsoever. This is a big problem and there is a realistic threat of massive fraud that will not be detected for months. A company can sell 1 million false tickets over the internet for flights taking off in 3 months, and the fraud will be visible only on the date of the flight, where many people will start showing up at the airports for flight that never existed and in the name of companies that maybe never existed either.

Electronic Airline Tickets are usually sent over the internet and users print these tickets. The ideal solution to this problem is to embed in these tickets a security code, that could be verified at the web site of some recognised authority, to be signed by one of the registered airlines. At the same time the user might get additional information: about strikes, flight delays, airport access, etc.

This security code should be short, in order for user to be able to read it from a printed electronic version, and type it in for verification. Using a digital signature for producing these security codes has a lot advantages to the user. For example, non-repudiation assures the user that the airline is aware of having emitted this ticket and there should be no tickets sold to several people. Public verifiability of digital signatures assures that disputes with customers can be decided and solved efficiently by a third party.

2. Serial numbers for software.

Off-line software activation scenario.

Most software vendors deliver it on a CD-ROM with a serial number that is printed as a series of symbols that has to be short (for example 40-80 bits), in order to be easily manipulated by the user (only very expensive software use hardware dongles). Some more or less obscure algorithm is used to produce and verify serial numbers.

With this kind of solution, as long as this algorithm is based on classical secret-key-type cryptographic techniques, the average hacker that disposes of the verification program for the serial numbers, can produce a key generator that works for this software. The key generator together with a copy of the original CD constitutes a perfect untraceable copy of the software. Key generators are currently produced and distributed on the internet at a massive scale and are driving thousands of small software developers out of business.

A much better method of generating serial numbers is based on short (public-key) signatures. By definition of a public-key solution, the description of the key generator cannot be derived from the verifier (the CD with the software). Now hackers have to modify the software itself in order to make a pirate copy of it. (The program may have hidden checks of integrity and such a modified copy will always be more traceable.)

On-line software activation scenario.

The public key method is also a big advantage in the on-line scenario, in which the user connects via a web page or via the telephone to the authorisation server of the software vendor, that will deliver him a serial number that will work only on one PC: a unique ID derived from the computer hardware and personal data is also signed by the software vendor. Here the size of the signature and of the unique ID matter a lot for people that want to use the telephone to register and activate their software.

3. Authorisation of Credit Cards.

In many countries for purchases with magnetic strip cards, the vendor or the vending machine connects via a phone line to the authorisation center that delivers an authorisation number for this transaction. For a vendor this number is very important and usually it guarantees him that he will be paid for the goods sold (otherwise the card might have been reported stolen just before the transaction). If this authorisation number that is short and transmitted by the phone were a digital signature, then the vendor would have a better confidence in it. In the present system, a thief in a very expensive jewellery shop may actually intercept the phone call and buy an expensive diamond with a stolen card.

4. Electronic post stamps.

The idea is to print post stamps at home with a picture of a new born baby, or print them inside a big company with an advertising capacity and a reduced cost of handling email for the postal services. This idea have been commercially developed by a californian start-up www.photo.stamps.com in agreement with US Postal Service. They sells stamps 1\$ each instead of regular price of 37 cents, and in spite of the price difference they claim to have sold more than 1.5 million of them already (read in the French newspaper Les Echos, 29 September 2004).

The authenticity of the stamp is provided by a digital signature that is machine-readable and printed on a surface of few square milli-meters (to save space for the picture). Therefore the signature have to be short. The stamps contain a signature by the electronic stamps authority of

- the name of the user that is authorised to emit/print them (it has an account open with the post office),
- and of the current day (and maybe even exact hour).

It is still possible to duplicate someone's stamps with a colour copier, but when we receive a letter with the stamp it is no longer valid. And the user can emit an unlimited number of stamps himself, it is electronic readers at the post office that will count them and sell him an invoice to pay at the end of the month.

5. Electronic Banking, Bank Notes, Printed Documents and Certificates.

Another big area of application of short signatures are all types of printed documents that are certificates and cannot legally be duplicated. In order to enforce this legal protection this type of documents do usually embed at a time many different protections against fraud: special paper and ink, special printing techniques, holograms, watermarking, embedding pieces of metal, etc. For many documents their authenticity

should look convincing even when we look at a photo-copy (for example ID-cards) yet parts of the documents may be printed in such a way that they will not be easy to photocopy and duplicate faithfully. We give here some examples:

- Bank notes: if a machine readable digital signature is a serial number of a bank note, its authenticity can be verified publicly and off-line with the exception of duplicating the same note, which is easy to trace.
- The same applies to cheques.
- Identity cards and passports may have a serial number that is a digital signature of some basic personal data of the person. Then the authenticity can be verified by fast machine readers at border checkpoints. Moreover, the authenticity of these documents can be verified by anyone, which is very useful in business transactions, administration and everyday life. For example a person that wants to rent a car, or an apartment, leaves a photocopy of his ID, and its authenticity can be verified at any moment, even from a photocopy.

This function of “authenticated photocopy” of current ID cards has always been used in practice on a large scale. Yet it is not addressed by ID cards and passports with embedded chips, and has to be strengthened in the time everybody is tempted to produce false documents with a home PC, and constantly improving scanners and printers.

The security given by this type protection is the undeniable proof that an ID card for this name was either delivered to somebody, or duplicated from one authentic card that identifies and allows to trace the frauds.

6. Short signatures can be used when it is necessary to sign every piece of data written in a smart card but yet there is a very small amount of memory available. We give two examples:

- Health card given to every citizen. A doctor writes/modifies small amounts of data (few bytes, for example prescribed drugs, the results of some examinations) in a card that has a small amount of non-volatile memory. Then he signs this data as things being prescribed/found/done by a certified practitioner. The signature cannot be long because the memory is scarce and data to be signed very short.
- In the current standards for bank cards with embedded microprocessor chips, there is an important protection that protects the user **even if** somebody would be able to duplicate both the magnetic stripe and the exact content of the tamper-resistant on-card chip.

If the user still has his card, and all expenses he has done in the last few months are written in the non-volatile memory that he cannot selectively erase. He can show these to the bank and repudiate all transactions that are not written in his card. In order to store as many transactions as possible in the scarce memory of the card (ideally all transactions for the lifetime of the card) and in order to prevent the user from claiming that a teller has written in his card many dummy transactions that never existed, the amounts of transactions written in the card should be signed (by both parties) at the moment of transaction.

As long as the user has his card, transactions can be repudiated even if the tamper resistance is breached. If the user loses his card, he should revoke it immediately.

2.1.2 Main Categories of Schemes and Their Particularities

First of all we should note that the mainstream digital signatures such as RSA does not provide the solution. Another remark is that, it is already very difficult to invent a new public key scheme, and for schemes with special properties such as short signatures (sometimes extremely short is demanded) it is even harder. As a consequence, most solutions known in the area of short signatures are not as well studied and understood as RSA, yet are interesting to study because if their security is lower than expected we learn new cryptanalytic techniques, and otherwise we solve difficult practical problems for which few solutions are known.

Acknowledgments

Contributors from ECRYPT: Nicolas Courtois, Louis Goubin, Louis Granboulan, Tanja Lange.

References

- [1] S. Goldwasser, S. Micali and R. L. Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*. SIAM Journal on Computing, 17(2):281-308, Apr. 1988.

2.2 Transitive signatures

2.2.1 The concept

Transitive signatures allow a signer to dynamically build an authenticated graph, edge by edge. The signer, having public key tpk and secret key tsk , can issue a signature on an (undirected) edge $\{i, j\}$, thereby adding $\{i, j\}$ to the graph. This signature can later be verified by anyone having the public key. Transitivity is realized by an additional composition algorithm that, given the public key and two signatures on adjacent edges $\{i, j\}$ and $\{j, k\}$, outputs a valid signature for the direct edge $\{i, k\}$. Hence, the authenticated graph does not only contain the edges explicitly signed by the signer, but comprises the entire transitive closure of these edges.

Military chains of command and administrative domains were mentioned as practical applications of transitive signatures [MR02], but a truly compelling application has yet to be found.

2.2.2 Realizing the concept

DEFINITIONS. Following the definitions of Micali and Rivest [MR02], a transitive signature scheme is a tuple of four polynomial-time algorithms $\mathcal{TS} = (\text{TKg}, \text{TSign}, \text{TVf}, \text{Comp})$. The key generation algorithm TKg , on input 1^k where $k \in \mathbb{N}$ is the security parameter, outputs a public key tpk and a matching secret key tsk . To add an edge $\{i, j\}$ to the authenticated graph, the signer runs the signing algorithm TSign on input the secret key tsk and nodes

i, j , producing a signature σ . Checking the validity of a candidate signature σ is done by the verification algorithm TVf when given the public key tpk , nodes i, j and σ as input. Finally, the composition algorithm Comp takes as input the public key tpk , nodes i, j, k , and signatures σ_1, σ_2 on edges $\{i, j\}$ and $\{j, k\}$. It produces a third signature σ_3 on edge $\{i, k\}$ as output. Security requires that an adversary who has adaptive access to a signing oracle cannot forge a signature on an edge that is not within the transitive closure of the edges signed by the oracle. Defining correctness is a bit tricky, as the general definition for homomorphic signatures [JMSW02] does not apply due to the statefulness of the signing algorithm of many schemes; a definition overcoming these difficulties was given in [BN02].

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 . The growth in signature size and the loss of privacy incurred by having signatures carry information about their history, however, may not be tolerable in certain situations, motivating the search for special-purpose schemes.

THE NODE CERTIFICATION PARADIGM. The main result of [MR02] is the first (non-trivial) transitive signature scheme, that is proven to be unforgeable under adaptive chosen-message attack assuming that the discrete logarithm problem is hard in an underlying prime-order group and assuming security of an underlying standard signature scheme. This scheme follows what Bellare and Neven [BN02] later called the *node certification paradigm*. The signer’s keys include those of a standard digital signature scheme, and the public key includes additional items. (In the scheme of [MR02], this is the description of a group of prime order q and a pair of generators of the group.) 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 signature scheme. The transitive 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 edge labels.

Following the same paradigm, Bellare and Neven [BN02, BN04] later introduced a number of new schemes based on alternative assumptions, being the one-more RSA assumption [BNPS03]¹, factoring, the one-more discrete logarithm assumption [BNPS03], and the one-more Gap Diffie-Hellman assumption [Bol03].

The paradigm is useful, but brings an associated cost. Producing a signature for an edge can involve computing two standard 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.

ELIMINATING NODE CERTIFICATES VIA HASHING. Bellare and Neven [BN02, BN04] observed that some schemes are 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 idea is that the public label of a node is not chosen by the signer, but rather implicitly specified as the output of a public hash function applied to the name of

¹This scheme was actually already presented by [MR02], but only proven secure against non-adaptive adversaries under the one-wayness of RSA. Bellare and Neven gave a proof against adaptive adversaries under the one-more RSA assumption.

the node. The signer’s secret key contains a piece of trapdoor information that allows him to compute the edge label from the public labels of the endpoints. More particularly, this technique can be applied to the schemes based on factoring, one-more RSA and one-more gap Diffie-Hellman. The security of these schemes is proven in the random oracle model [BR93].

2.2.3 Open problems

DIRECTED TRANSITIVE SIGNATURES. All transitive signature schemes known today are for undirected graphs only. If truly compelling applications of transitive signatures exist, they are more likely to be found for directed graphs than for undirected ones. At this point, no constructions for directed transitive signatures have been proposed, and Hohenberger [Hoh03] even provided evidence that they may be very hard to construct. She 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, of which no examples are 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 functionality 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. As an application of transitive signatures, one may think of shrinking so-called certificate chains to a single signature. Certificate chains are used to trace the authenticity of a user’s public key back to a root certificate that is typically embedded in the verifiers software. They arise from hierarchically structured public key infrastructures, in which each certification authority (CA) signs the public key of the next. A certificate chain of length n tracing a user’s public key pk_n back to a root public key pk_0 contains n signatures and n public keys, as follows:

$$pk_n \parallel \text{Sign}(sk_{n-1}, pk_n) \parallel pk_{n-1} \parallel \text{Sign}(sk_{n-2}, pk_{n-1}) \parallel \dots \parallel pk_1 \parallel \text{Sign}(sk_0, pk_1) .$$

In spite of the first-sight analogy between graphs and certification trees, transitive signatures cannot help to compress this chain into a single signature, because 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, LMRS04] 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

$$pk_n \parallel pk_{n-1} \parallel \dots \parallel pk_1 \parallel \sigma .$$

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. What we need is 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.

Acknowledgments

Contributors from ECRYPT: Gregory Neven.

References

- [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, Berlin Germany, 2003.
- [BN02] 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, Berlin Germany, 2002.
- [BN04] Mihir Bellare and Gregory Neven. Transitive signatures: New schemes and proofs. Cryptology ePrint Archive, Report 2004/215, 2004. <http://eprint.iacr.org/>.
- [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.
- [Bol03] 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, Berlin Germany, 2003.
- [BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.
- [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, Berlin Germany, 2002.
- [LMRS04] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin, and Hovav Shacham. Sequential aggregate signatures from trapdoor permutations, 2004.

- [MR02] 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, Berlin Germany, 2002.

2.3 Aggregate signatures

2.3.1 Introduction

The concept of *aggregate signatures* was introduced by Boneh, Gentry, Lynn and Shacham [2] at Eurocrypt 2003. An aggregate signature scheme is a digital signature that supports aggregation: given k signatures on k distinct messages from k different users it is possible to aggregate all these signatures into a single short signature. This useful primitive allows to drastically reduce the size of public-key certificates, thereby saving storage and transmission bandwidth.

2.3.2 Notations

We will adopt [2]’s notations and settings, namely:

- G_1 and G_2 are two multiplicative cyclic groups of prime order p ;
- g_1 is a generator of G_1 and g_2 is a generator of G_2 ;
- ψ is a computable isomorphism from G_1 to G_2 with $\psi(g_1) = g_2$;
- e is a computable bilinear map $e : G_1 \times G_2 \rightarrow G_T$ where G_T is multiplicative and of order p . The map e is:
 - Bilinear: for all $u \in G_1, v \in G_2$ and $a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$
 - Non-degenerate: $e(g_1, g_2) \neq 1$
- $h : \{0, 1\}^* \rightarrow G_2$ is a hash function.

Figure 1 briefly recalls Boneh, Lynn and Shacham’s signature scheme [1], upon which the aggregate signatures schemes of [2, 3] are based.

2.3.3 Aggregate Signatures

Consider now a set of k users using Figure 1’s scheme (each user having a different key pair bearing an index i) and signing different messages M_i . Aggregation consists in combining the resulting k signatures $\{\sigma_1, \dots, \sigma_k\}$ into one aggregate signature σ . This is done by simply computing:

$$\sigma \leftarrow \prod_{i=1}^k \sigma_i$$

Key generation	
	Pick random $x \xleftarrow{R} \mathbb{Z}_p$
	Compute $v \leftarrow g_1^x$
Public :	$v \in G_1$
Private :	$x \in \mathbb{Z}_p$
Signature	
	Hash the message $M \in \{0,1\}^*$ into $h \leftarrow h(M) \in G_2$
	Compute the signature $\sigma \leftarrow h^x \in G_2$
Verification of σ (with respect to v and M)	
	Compute $h \leftarrow h(M)$
	Check that $e(g_1, \sigma) = e(v, h)$

Figure 1: Boneh, Lynn, Shacham Signatures

Aggregate verification is very simple and consists in checking that the M_i are mutually distinct and ensuring that:

$$e(g_1, \sigma) = \prod_{i=1}^k e(v_i, h_i) \quad \text{where } h_i = h(M_i)$$

This holds because:

$$e(g_1, \sigma) = e(g_1, \prod_{i=1}^k h_i^{x_i}) = \prod_{i=1}^k e(g_1, h_i)^{x_i} = \prod_{i=1}^k e(g_1^{x_i}, h_i) = \prod_{i=1}^k e(v_i, h_i)$$

Acknowledgments

Contributors from ECRYPT: Jean-Sébastien Coron.

References

- [1] D. Boneh, B. Lynn and H. Shacham, *Short Signatures From the Weil Pairing*, Proceedings of ASIACRYPT' 2001, Lecture Notes in Computer Science vol. 2248, Springer-Verlag, pp. 514-532, 2001.
- [2] D. Boneh, C. Gentry, B. Lynn and H. Shacham, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps*, Advances in Cryptology - EUROCRYPT' 2003 Proceedings, Lecture Notes in Computer Science vol. 2656, E. Biham ed., Springer-Verlag, pp. 416-432, 2003.
- [3] D. Boneh, C. Gentry, B. Lynn and H. Shacham, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps*, Cryptology ePrint Archive, Report 2002/175, 2002, <http://eprint.iacr.org/>.

2.4 Fail-stop signatures

In this section, we call signatures as introduced in [3] and formally defined in [4], i.e., signatures without special properties, “ordinary” digital signatures for distinction.

2.4.1 Introduction

Ordinary digital signatures allow a person, A (for Alice), to make signatures that everybody knowing A 's public key can test. Such signatures are only computationally secure for the signer, because they can be forged by persons with sufficiently large computing power. A person able to factor large integers can, for example, very easily forge RSA signatures. Hence the security of these schemes relies on a computational assumption. Moreover, if a signature should be forged, it will be difficult for A to convince the bearer of the signed document or a third party that she did not make that signature.

Fail-stop signatures solve this problem by offering the signer a method for proving that a forgery has taken place. More precisely, even if a forger with unlimited computing power makes an “optimal” forgery, a polynomially bounded signer can prove that the underlying computational assumption has been broken when she sees the forgery (except with negligible probability). Thus the signer can be protected from an arbitrarily powerful forger. Moreover, after the first forgery, all participants, or the system operator, know that the signature scheme has been broken, so that it can be stopped. Hence the name “fail-stop”.

More About Ordinary Digital Signatures. For ordinary signature schemes of the original structure, it was already shown in [3] that the security can only be computational: The signer has a secret key, which she uses to make signatures, and the signatures can be tested by everyone knowing her corresponding public key. As signing and testing are polynomial-time, one can forge signatures, i.e., find values that pass the test, in nondeterministic polynomial time simply by guessing among all values up to a certain length. Additionally, non-polynomial lower bounds for problems within NP are not known; hence to prove the security of any ordinary digital signature scheme, one has to make a computational assumption. The same argument also applies to the more general construction in [4].

More About the Fail-Stop Property. The fail-stop property can be described best by considering a judge in a dispute between the signer and a recipient of a digital signature. Usually, the judge will test if the signature is correct and give his verdict “ok” or “not ok” accordingly. Fail-stop signatures supply the judge with a third possibility: If the signer can prove that the signature is forged, the judge may say “forgery proved”, which can be interpreted as saying that the basic assumption of the system has been broken. Naturally, this possibility of distinguishing forged signatures from authentic signatures only exists as long as the forger has not stolen the signer's key.

The definition of fail-stop signatures does not specify for how much of a system a particular proof of forgery is valid. As long as forging a single signature is provably as hard as breaking a particular computational assumption, it is wise to stop the whole system after any forgery, because if one signature could be forged, one must expect that the same forger can make more

forgeries. Therefore, the constructions usually assume that there is only one type of proof of forgery. However, it is no problem to make proofs of forgery specific to the keys of individual signers or even (although currently with some loss in efficiency) to each particular signature.

Furthermore, it is not a matter of the definition how one acts after the output “forgery proved”. In particular, one obtains exactly the properties of ordinary digital signatures again if the technical verdict “forgery proved” is interpreted just like “signature correct” by the judge and everybody else. On the other hand, if it is agreed that all signatures for which forgery can be proved are rejected, one obtains a signature scheme in which the signer is unconditionally protected against forgeries, whereas the recipient is only computationally protected, i.e., an unrestricted adversary may achieve that a signature accepted by the recipient is later rejected by an honest judge. We call such a signature scheme a “dual signature scheme”, because it is dual to ordinary digital signatures with respect to the security for the signer and recipients. As fail-stop signatures furthermore allow the system to be stopped as soon as the basic assumption has been broken, they are a strictly stronger notion than each of these types of signatures.

As an example where fail-stop signatures, in their special role as dual signatures, may be advantageous, consider an electronic payment system where a customer signs her requests to the bank digitally. As the bank, most likely, has much more computing power than the customer, and as it can select the system and choose the security parameters, it is reasonable to protect the customer unconditionally, whereas the bank can rely on the customer not having sufficient computing power to repudiate her signatures. Thus the customer should sign with dual signatures and the bank with ordinary digital signatures.

2.4.2 Description

Briefly, an ordinary digital signature scheme is defined by three algorithms:

1. a key generator,
2. a method for signing, and
3. a method for testing signatures,

such that if the keys are generated correctly using the key generator, then:

- If the signer signs a message correctly, everybody knowing the signer’s public key accepts the signature.
- A polynomially bounded forger cannot make any signature that passes the signature test.

In a fail-stop signature scheme, a protocol is added that allows the (polynomially bounded) signer to prove to third parties that a forged signature is indeed a forgery. It consists of two more algorithms:

1. a method for constructing proofs of forgery, and

2. a method for verifying proofs of forgery (which everybody knowing the public key can carry out).

A proof of forgery is always non-interactive, so that it can subsequently be shown to others, and the system can be stopped in consensus. The proof must satisfy two new security requirements:

- The ability to prove forgeries must work independently of the computing power of potential forgers.
- It must be infeasible for the signer to construct signatures that she can later prove to be forgeries.

It can be shown that these two properties imply security against forgery.

Until now, nothing has been said about the generation of the secret and public key for fail-stop signatures. In an ordinary digital signature scheme, the primary purpose of the secret key is to enable the signer to make signatures that nobody else can construct, and this key should therefore be chosen by the signer. As it is equally important that fail-stop signatures cannot be forged, the signer still has to take part in choosing the keys. However, as the signer in these schemes is allowed to repudiate (forged) signatures despite the fact that they pass the public signature test, the recipients of signatures must be sure that the signer cannot disavow her own signatures. It is therefore necessary that the recipients or a center trusted by the recipients also participate in the key generation. In particular, such a center is needed if the recipients are not known at the time of choosing the keys. When the signer's public key has been selected, it will usually be stored in a public directory with accepted integrity. For the detailed formal definition of fail-stop signatures, both the algorithms and the security properties, we refer to [5].

2.4.3 State of the Art

Fail-stop signatures were introduced around 1990, mainly in [7, 8]. The status in 1996 is summarized in [5, 6]. Later papers are closely linked to the overall constructions from these papers.

Overall Constructions. Overall constructions of fail-stop signature schemes work in three steps:

- First one constructs a one-time fail-stop signature scheme for a message of bounded length.
- To sign messages of unbounded length, one uses a hash-then-sign scheme with a collision-resistant hash function. Hash collisions count as proofs of forgery.
- To sign multiple messages, one uses a tree or chaining construction on top of the one-time system. There are four main variants:

- Bottom-up trees. Here the public keys for multiple one-time fail-stop signatures are hashed in a hash tree. Hash collisions count as proofs of forgery.
- Top-down trees. Here each node of the tree contains a key pair, and the secret key in each inner node is used to sign the public keys of its children.
- Accumulators. This is similar to bottom-up trees, but a special primitive called accumulator allows the omission of a tree structure [1].
- If a signer only sends signatures to one recipient (e.g., in the bank example above), one can use simpler and more efficient chains instead of trees.

If one trusts the collision-resistance of fast standard hash functions, the bottom-up trees are the most efficient variant for arbitrary recipients, while bottom-up trees are somewhat more flexible and allow a smaller amount of secret storage in some cases, and accumulators yield shorter signatures.

One-time Fail-stop Signature Constructions. Most constructions of the basic one-time fail-stop signatures rely on families of bundling homomorphisms. These are collision-resistant group homomorphisms with the additional property that all images have large preimage sets. Given a homomorphism h from such a family, a secret key is a pair (sk_1, sk_2) of preimages, the public key is the pair of images $pk_i := h(sk_i)$, and the signature of a message m from the preimage group is

$$sig := sk_1 + m \cdot sk_2.$$

A signature is valid iff

$$h(sig) = pk_1 \cdot pk_2^m.$$

A proof of forgery is simply a pair of two different signatures for the same message. It is easy to see that this yields a correct scheme and that constructing a proof of forgery is equivalent to finding a collision of the bundling homomorphism. The probability that the signer can indeed find a proof of forgery for an arbitrary forged signature depends on the bundling degree of the homomorphism and the group structure, see [5].

Bundling homomorphisms can be constructed based on discrete logarithm assumption (simply the function that maps pairs (x, y) to $g^x h^y$ for certain given groups and generators g and h), and less obviously based on the factoring assumption [5]. Some later papers present variations of this concept [13, 12, 11, 10]. Not all of them are still considered secure [10].

There are also combinations of the fail-stop property and other special signature properties such as undeniable or threshold.

2.4.4 Open Problems

The main efficiency difference between fail-stop and ordinary signature schemes lies in the tree or accumulator constructions used for signing multiple messages. It is proven that the amount of secret randomness needed by the signer in fail-stop signatures grows linearly in the number of signatures [5]. This is an indication that something like a tree or accumulator construction may be needed, but not a proof that the known constructions based on clearly separated one-time key pairs are the only possible ones or optimal.

On the theoretical side, ordinary signature schemes are known to exist if any one-way function exists [9], while fail-stop signatures are only known to exist if either one-way permutations or collision-free hash functions exist [2].

Fail-stop signatures have not been adopted in real life. However, even ordinary signature schemes have almost no such adoption, at least not for non-repudiation. This can be attributed to several factors. One is the lack of secure “what you see is what you sign” solutions for workstations, which make non-repudiation very dubious in practice, even if signature regulations tend to ascribe it to implementations that do not really offer this feature. Another factor is the lack of public-key infrastructures beyond individual enterprises, and the lack of a business case with clear and fair liability distribution for them. This reduces potential signature usage to situations where cryptographic non-repudiation plays no role. Hence the main motivation for fail-stop signatures, the situation with a judge, does not exist in real life yet. If signatures become much more widely deployed, and weaker links in the chain have been addressed, the question of reliably detecting cryptographic forgeries may well find interest. Then fail-stop signatures will become relevant.

Acknowledgments

Contributors from ECRYPT: Birgit Pfitzmann.

References

- [1] Niko Barić, Birgit Pfitzmann: Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees; Eurocrypt '97, LNCS 1233, Springer-Verlag, Berlin 1997, 480-494.
- [2] Ivan B. Damgård, Torben P. Pedersen, Birgit Pfitzmann: On the Existence of Statistically Hiding Bit Commitment Schemes and Fail-Stop Signatures; Journal of Cryptology 10/3 (1997) 163-194.
- [3] Whitfield Diffie, Martin E. Hellman: New Directions in Cryptography; IEEE Transactions on Information Theory 22/6 (1976) 644-654.
- [4] Shafi Goldwasser, Silvio Micali, Ronald L. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks; SIAM Journal on Computing 17/2 (1988) 281-308.
- [5] Torben P. Pedersen, Birgit Pfitzmann: Fail-stop Signatures; SIAM Journal on Computing 26/2 (1997) 291-330.
- [6] Birgit Pfitzmann: Digital Signature Schemes – General Framework and Fail-Stop Signatures; LNCS 1100, Springer-Verlag, Berlin 1996.
- [7] Birgit Pfitzmann, Michael Waidner: Formal Aspects of Fail-stop Signatures; Interner Bericht 22/90, Fakultät für Informatik, Universität Karlsruhe, 1990.

- [8] Birgit Pfitzmann, Michael Waidner: Fail-stop Signatures and their Application; 9th Worldwide Congress on Computer and Communications Security and Protection (Securicom 91), Paris, March 1991, 145-160.
- [9] John Rompel: One-Way Functions are Necessary and Sufficient for Secure Signatures; 22nd Symposium on Theory of Computing (STOC), ACM, New York 1990, 387-394.
- [10] Katja Schmidt-Samoa: Factorization-based Fail-Stop Signatures Revisited; to appear in Information and Communications Security (ICICS 2004), Springer-Verlag, 2004.
- [11] Willy Susilo, Rei Safavi-Naini: An Efficient Fail-stop Signature Scheme based on Factorization; Information Security and Cryptology (ICISC 2002), LNCS 2587, Springer-Verlag, Berlin 2003, 62-74.
- [12] Willy Susilo, Rei Safavi-Naini, Marc Gysin, Jennifer Seberry: A New and Efficient Fail-Stop Signature Scheme; The Computer Journal 43/5 (2000) 430-437.
- [13] Willy Susilo, Rei Safavi-Naini, Josef Pieprzyk: RSA-based fail-stop signature schemes; International Workshop on Security (IWSEC'99), IEEE Computer Society Press, 1999, 161-166.

2.5 Key-evolving signatures

2.5.1 The key-exposure problem

In a vast majority of existing cryptographic solutions, security guarantees last only as long as secrets remain unrevealed. If a secret is revealed (either accidentally or via an attack), security is often compromised not only for subsequent uses of the secret, but also for prior ones. For example, if a secret signing key becomes known to an adversary, one cannot trust any signature produced with that key, regardless of when; if a secret decryption key becomes known to an adversary, then any encrypted message, even if sent long before, is not guaranteed to remain private.

To address this problem, several different approaches have been suggested. Many attempt to lower the chance of exposure of secrets by distributing them across several systems, usually via secret sharing. As pointed out in [BM99], this method is usually quite costly, and may, in fact, be too expensive to be implemented by a typical individual user. Moreover, since each of the systems may be susceptible to the same attack, the actual risk may not decrease.

A complementary approach is to reduce the potential damage in case secrets are exposed. The main idea in this case is to ensure that secrets are used only for short time periods, and that compromise of a secret does not affect anything based on secrets from other time periods. One of the challenges in designing such a system is to be able to change secret information without the inconvenience of changing public information, such as the public key. Two main lines of work in this direction are *forward-secure signatures* and *key-insulated signatures*

2.5.2 Forward-secure signature schemes

The idea of forward-secure signature schemes was proposed by Anderson in [And00] and formalized by Bellare and Miner in [BM99]. Informally, a forward-secure signature scheme is a *key-evolving scheme* whose operation is divided into time periods, with a different secret key for each time period. Each secret key is used to sign messages only during a particular time-period, and to compute a new secret key at the end of that time period. It is then erased. As in ordinary signature schemes, however, there is only one public key, which remains the same through all the time periods. The verification algorithm checks not only that a signature is valid, but also that it was generated during a specific time period.

Such a scheme is *forward-secure* if it is infeasible for an adaptive chosen-message adversary to forge signatures for past time periods, even if it discovers the secret key for the current time period. Note that, in particular, this implies that past secret keys cannot be recovered from the current one. In a forward-secure signature scheme, even if the current secret key is compromised, signatures from past time periods can still be trusted.

Definition A forward-secure digital signature scheme is, first of all, a key-evolving digital signature scheme. A key-evolving signature scheme is very similar to a standard one. Like a standard signature scheme, it contains a key generation algorithm, a signing algorithm, and a verification algorithm. The public key is left unchanged throughout the lifetime of the scheme, making the verification algorithm very similar to that of a standard signature scheme. Unlike a standard signature scheme, a key-evolving signature scheme has its operation divided into time periods, each of which uses a different (but related) secret key to sign a message. The way these keys are updated is given by a public update algorithm, which computes the secret key for the new time period based on that for the previous one. The forward security comes, in part, from the fact that this update function is one-way and, given the secret key for the current period, it is hard to compute any of the previously used secret keys. It is important, of course, for the signer to delete the old secret key as soon as the new one is generated, since otherwise an adversary breaking the system could easily get hold of these undeleted keys and forge messages for time periods preceding that of the break-in.

Security The formal security model for key-evolving signature schemes was introduced by Bellare and Miner [BM99], which extends that of Goldwasser, Micali and Rivest [GMR88] to take into account the ability of the adversary to obtain a key by means of a break-in. In this model, besides knowing the user's public key PK , the adversary also gets to know the total number of time periods and the current time period. The adversary runs in three phases. In the first phase, the chosen message attack phase (**cma**), the adversary has access to a signing oracle, which it can query to obtain signatures of messages of its choice with respect to the current secret key. At the end of each time period, the adversary can choose whether to stay in the same phase or switch to the break-in phase (**breakin**). In the break-in phase, which models the possibility of a key exposure, we give the adversary the secret key SK_j for the specific time period j it decided to break in. In the last phase, the forgery phase (**forge**), the adversary outputs a pair signature-message, that is, a forgery. The adversary is considered to be successful if it forges a signature of some new message (that is, not previously queried to the signing oracle) for any time period prior to j .

Constructions In the literature, one can find two main classes of forward-secure signature schemes: generic and non-generic ones. While generic forward-secure signature schemes can be built from any ordinary signature scheme, non-generic ones are usually based on very specific signature schemes.

GENERIC CONSTRUCTIONS. The first generic construction of a forward-secure signature scheme whose parameters do not all grow linearly with the number of time periods, was proposed by Bellare and Miner in [BM99]. Their generic construction, which involves the use of binary-tree-based certification chains, yields schemes for which the public key length is independent of the number of time periods T and for which secret keys and signatures are of size linear in $\lg(T)$. The efficiency of the key generation and key update algorithms is also linear in $\lg(T)$, but their efficiency can be made independent of T by using a technique due to Canetti *et al.* [CHK03] in which all the nodes of the tree are associated with a time period (in a pre-order traversal) instead of the leaves only.

Other important generic constructions are the ones of Krawczyk [Kra00] and Malkin *et al.* [MMM02]. In [Kra00], a forward-secure signature consists of only 2 ordinary signatures, but his construction requires the storage of non-secret certificates of size linear in T . In [MMM02], Malkin *et al.* describe general techniques to build forward-secure signature schemes for which the efficiency and security depend only on the number of time periods elapsed so far. In their construction, the number of time periods does not need be known in advance and is not part of the public key.

NON-GENERIC CONSTRUCTIONS. The first non-generic construction of a forward-secure signature scheme was proposed by Bellare and Miner in [BM99], using ideas from the Fiat-Shamir [17] and Ong-Schnorr [24] identification schemes. Their construction, which is proven forward-secure in the random oracle model of [BR93] based on the hardness of factoring, relies heavily on a key feature of these identification schemes: the prover does not need to know the factorization of the modulus, but only modular square roots of some public numbers. The key update in this case consists simply of squaring computations, which is a one-way function in this setting.

Another important non-generic construction is that of Abdalla and Reyzin in [AR00], in which they propose a forward-secure signature scheme based on the 2^t -root scheme [Mic94, OO90, 24]. Their scheme, in comparison to that of Bellare and Miner [BM99], possesses shorter keys, but is less efficient in terms of signing and verifying. Their proof of security is also in the random oracle model and based on the hardness of factoring. While requiring less space than the generic schemes, both the Bellare-Miner [BM99] and the Abdalla-Reyzin [AR00] schemes require signing and verification times that are linear in T .

The state of the art is the construction of Itkis and Reyzin in [IR01], in which they proposed the first forward-secure signature scheme for which both signing and verifying are as efficient as for the Guillou-Quisquater signature scheme [GQ90], one of the most efficient ordinary signature schemes. Their scheme is provably secure in the random oracle model based on a variant of the strong RSA assumption.

2.5.3 Key-insulated signature schemes

The concept of key-insulated signature schemes was first formalized by Dodis *et al.* in [DKXY03] based on similar concepts from [DKXY02] for public-key encryption. As forward-secure signature schemes, a key-insulated scheme is also a *key-evolving scheme* whose operation is divided into time periods, with a different secret key for each time period and with each secret key being used to sign messages only with respect to its particular time-period. The main difference between the two, however, is that the key update is done with the help of an additional secret key, called the *master key*, which is stored in a physically secure device (i.e., tamper-resistant).

The security requirements of key-insulated signature schemes differ from those of forward-secure signature schemes. More specifically, in a key-insulated scheme, the adversary may be able to compromise keys for several different time periods, which are not necessarily consecutive. A scheme is then said to be *key-insulated* if it is infeasible for an adaptive chosen-message adversary to forge signatures for any time period for which this adversary has not obtained the associated secret key.

STRONG KEY-INSULATED SIGNATURES. In [DKXY03], Dodis *et al.* also consider cases in which the device holding the master key is not fully trusted. In such situations, the actual secret key that is used to sign messages should not be computable only from the master key, but rather from both the master key and the secret signing key of the current time period. In fact, the master key is only used in this case to generate a partial key which can then be used by the signing device to compute the actual signing key.

The security definitions for strong key-insulated signatures are similar to those for ordinary key-insulated ones, except that the adversary is only allowed to corrupt either the signing device or the device holding the master key, but not both.

CONSTRUCTIONS. In [DKXY03], Dodis *et al.* proposed several constructions of strong key-insulated signatures. The first of these is a generic construction of a strong key-insulated signature scheme from any ordinary signature scheme. The security in this case relies solely on that of the underlying signature scheme and can withstand any number of corruptions of the signing device.

The second construction proposed by Dodis *et al.* is a discrete-logarithm-based one which can withstand up to t corruptions of the signing device, where t is a parameter of the scheme. The signing and the verification algorithms are more efficient than those of the generic construction, but the key update is linear in t . The proof of security is in the random oracle model and based on the hardness of the discrete logarithm problem. Their third and most efficient construction is based on specific types of ordinary signature schemes, which they call trapdoor signature schemes, and it can withstand any number of corruptions of the signing device. In the case where the ordinary signature scheme is the Guillou-Quisquater signature scheme [GQ90], the resulting key-insulated signature scheme can be proven secure in the random oracle model based on the RSA assumption.

Acknowledgments

Contributors from ECRYPT: Michel Abdalla.

References

- [And00] Ross Anderson. Two remarks on public-key cryptology. Manuscript. Relevant material presented by the author in an invited lecture at the ACM CCS 97: 4th Conference on Computer and Communications Security, Zurich, Switzerland, April 1–4, 1997, September 2000.
- [AR00] Michel Abdalla and Leonid Reyzin. A new forward-secure digital signature scheme. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 116–129, Kyoto, Japan, December 3–7, 2000. Springer-Verlag, Berlin, Germany.
- [BM99] Mihir Bellare and Sara Miner. A forward-secure digital signature scheme. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO’99*, volume 1666 of *Lecture Notes in Computer Science*, pages 431–448, Santa Barbara, CA, USA, August 15–19, 1999. Springer-Verlag, Berlin, Germany.
- [BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.
- [CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany.
- [DKXY02] Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung. Key-insulated public key cryptosystems. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 65–82, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer-Verlag, Berlin, Germany.
- [DKXY03] Yevgeniy Dodis, Jonathan Katz, Shouhuai Xu, and Moti Yung. Strong key-insulated signature schemes. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 130–144, Miami, USA, January 6–8, 2003. Springer-Verlag, Berlin, Germany.
- [FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO’86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer-Verlag, Berlin, Germany.

- [GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.
- [GQ90] Louis C. Guillou and Jean-Jacques Quisquater. A “paradoxical” indentity-based signature scheme resulting from zero-knowledge. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO’88*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231, Santa Barbara, CA, USA, August 21–25, 1990. Springer-Verlag, Berlin, Germany.
- [IR01] Gene Itkis and Leonid Reyzin. Forward-secure signatures with optimal signing and verifying. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 332–354, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany.
- [Kra00] Hugo Krawczyk. Simple forward-secure signatures from any signature scheme. In *ACM CCS 00: 7th Conference on Computer and Communications Security*, pages 108–115, Athens, Greece, November 1–4, 2000. ACM Press.
- [Mic94] Silvio Micali. A secure and efficient digital signature algorithm. Technical Memo MIT/LCS/TM-501b, Massachusetts Institute of Technology, Laboratory for Computer Science, April 1994.
- [MMM02] Tal Malkin, Daniele Micciancio, and Sara Miner. Efficient generic forward-secure signatures with an unbounded number of time periods. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 400–417, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer-Verlag, Berlin, Germany.
- [OO90] Kazuo Ohta and Tatsuaki Okamoto. A modification of the Fiat-Shamir scheme. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO’88*, volume 403 of *Lecture Notes in Computer Science*, pages 232–243, Santa Barbara, CA, USA, August 21–25, 1990. Springer-Verlag, Berlin, Germany.
- [OS90] H. Ong and Claus-Peter Schnorr. Fast signature generation with a Fiat Shamir-like scheme. In Ivan Damgård, editor, *Advances in Cryptology – EUROCRYPT’90*, volume 473 of *Lecture Notes in Computer Science*, pages 432–440, Aarhus, Denmark, May 21–24, 1990. Springer-Verlag, Berlin, Germany.

2.6 Blind signatures

2.6.1 Motivation: Electronic Cash

As early as 1982, Chaum’s [12] pioneering work aimed at creating an electronic version of money. To achieve this goal, he introduced the notions of “coins” and “randomized blind signatures” (or simply “blind signatures”). He claimed that this was the only way to ensure the required *anonymity*: in real life, a coin cannot be easily traced from the bank to the shop, furthermore, two spendings of a same user cannot be linked together. These are two main properties of real coins that Chaum wanted to mimic: *untraceability* and *unlinkability*.

He proposed to define an *electronic coin* as a number with a certificate (a signature) produced by the bank; it is withdrawn from the bank, spent by the user, and deposited by the shop.

On-line electronic cash. In his first scheme, Chaum used blind signatures for the production of coins. The user makes the bank blindly sign a coin. Then the user is in possession of a valid coin that the bank itself cannot recognize nor link with the user. When the user spends the coin, the shop immediately returns it to the bank. If the coin has already been spent, the bank detects the fact and informs the shop so that it refuses payment. It is an “on-line” context: there is a continuous communication between the shop and the bank in order to verify the validity of coins. In order to define the scheme, Chaum introduced the first blind signature scheme, based on the RSA hypothesis. It is a by now classical transformation of the original RSA signature scheme [30]:

The Blind RSA Signature. The bank has a large composite number $N = pq$, a public key e , and a related secret key d . It also uses a public hash function H . In order to get the signature of a random number ρ , the user “blinds” it with a random value $r^e \bmod N$, and sends $m = H(\rho)r^e \bmod N$ to the signer. The latter returns a signature σ' of m such that $\sigma'^e = m = r^e H(\rho) \bmod N$. Then the user can “unblind” this signature computing $\sigma = \sigma' r^{-1} \bmod N$. A coin is any pair (ρ, σ) which satisfies $\sigma^e = H(\rho) \bmod N$.

In this scheme all coins have the same value, but in a real system different denominations might be encoded by different exponents e .

Off-line electronic cash and the “cut-and-choose” methodology. In an “off-line” context we cannot prevent a user from spending a coin twice or even more, since the detection is made too late to refuse payment. This fraud is called “double-spending.” We only can hope that the double-spender will be discovered later and punished. Chaum et al. [13] were able to build such schemes by introducing the identity of the user in the coin in such a way that it remains concealed, unless double-spending happens. Once, blind signatures were a critical point for anonymity, and, as before, the authors used the blind RSA signature, together with the “cut-and-choose” technique: in their proposition, a coin is a kind of list of k blind signatures, each having an embedded copy of the identity of the user. To be sure that double-spending will reveal the real identity of the user, the bank would like to verify that the signatures actually have the requested format, which would revoke anonymity. Then the bank helps the user to get $2k$ signatures, randomly chooses k of them, and verifies the inner structure of the selected signatures. Since these signatures are no longer anonymous, the user throws them away and constructs the coin with the k other ones. The probability for a cheater to be finally in possession of a fraudulent coin is about 2^{-2k} .

The main drawback of the “cut-and-choose” technique is that the coins are very large, as well as the amount of computations. In 1993 Ferguson [16] and Brands [7] proposed new schemes without “cut-and-choose.” The first one uses once again the blind RSA signature, whereas Brands’ scheme uses a new blind signature derived from the Schnorr signature scheme [31], [32]:

The Blind Schnorr Signature. The generation algorithm produces two large prime integers p and q such that $q | p - 1$ as well as an element g of \mathbf{Z}_p^* of order q . It also creates

a pair of keys, (x, y) , where $x \in \mathbb{Z}_q^*$ is the secret one, and $y = g^{-x} \bmod p$ is the public one. The signer publishes y . In order to get the signature of a secret message m , the user asks the signer to initiate a communication. He chooses a random $K \in \mathbb{Z}_q^*$, computes and sends the “commitment” $r = g^K \bmod p$. The user then blinds this value with two random elements $\alpha, \beta \in \mathbb{Z}_q$, into $r' = rg^{-\alpha}y^{-\beta} \bmod p$, computes the value $e' = H(m, r') \bmod q$ and sends the “challenge” $e = e' + \beta \bmod q$ to the signer who returns the value s such that $g^s y^e = r \bmod p$. Finally, the user computes $s' = s - \alpha \bmod q$. This way, the pair (e', s') is a valid Schnorr signature of m since it satisfies $e' = H(m, g^{s'} y^{e'} \bmod p)$.

In both schemes Ferguson and Brands managed to hide the identity of the user in a much more efficient way than the “cut-and-choose” methodology. Again, the identity is revealed after double-spending. Those blind signatures which hide a specific structure, such as the identity, are called “restrictive blind signatures” [11], [9], [8], [29]. Many extensions [15], [6], [10] have been proposed, followed by some attacks [8], [11] and repairs [9], [34]. All of them use blind signatures, and the security of the proposed schemes is totally dependent on the security of the blind signatures they use.

2.6.2 Security Notions

However, no formal notion of security had ever been studied, or proved at that time, in the context of blind signatures. Nevertheless, it is a critical point in electronic cash systems. In the context of blind signatures, the previous definitions of security for signatures are no longer significant. In fact, existential forgery is somehow the basis for blind signatures. But a fundamental property for electronic cash systems is the guarantee that a user cannot forge more coins than the bank gives him. In other words, with ℓ blind signatures of the Bank, the user must not be able to create more than ℓ coins. This form of security was more or less informally assumed in connection with several schemes, for example in [10], or under the “unexpandability” property of [18]. The following security notions, which are now the classical ones, were defined in [28]:

Definition 2.1 [The $(\ell, \ell + 1)$ -Forgery] For any integer ℓ , an $(\ell, \ell + 1)$ -forgery comes from an attacker that produces $\ell + 1$ signatures after ℓ interactions with the signer $\{0, 1\}$.

Definition 2.2 [The “One-More” Forgery] For some integer ℓ , polynomial in the security parameter k , an attacker can obtain $\ell + 1$ valid signatures after fewer than ℓ interactions with the signer. In other words, a “one-more forgery” is an $(\ell, \ell + 1)$ -forgery for some polynomially bounded integer ℓ .

Definition 2.3 [The Strong “One-More” Forgery] An $(\ell, \ell + 1)$ -forgery for a polylogarithmically bounded integer ℓ (i.e., for some constant α , $\ell \leq (\log k)^\alpha$, where k is the security parameter) is called a *strong “one-more” forgery*.

As usual, several scenarios can be envisioned. We focus on two kinds of attacks which naturally come from the use of blind signatures in electronic cash :

- The *sequential attack*: the attacker interacts sequentially with the signer. This attack can be performed by a user who withdraws coins, one after the other.

It is clear that, in practical situations, many users might be allowed to withdraw money at the same time. The following attack must then be considered.

- The *parallel attack*: the attacker interacts ℓ times in parallel with the signer. This attack is stronger. Indeed, the attacker can initiate new interactions with the signer before previous ones have ended. This attack can be performed by a group of users who withdraw many coins at the same time.

By the way, beyond these unforgeability security notions, blindness must also be satisfied: given a signature, the signer should not be able to distinguish with whom the signer interacted to make it. In the two examples given above, the view of the signer is perfectly independent to the signature eventually obtained by the user, in an information-theoretical point of view. Therefore, the “blindness” property is perfect. More recently, “computational blindness” has been introduced [1].

2.6.3 Security Results

In [25, 28], a general description for blind signatures is given, so that the “forking lemma” technique [26] applies. They also propose several schemes which security is relative to the discrete logarithm problem or to RSA.

Witness Indistinguishability Previous methods of proofs used to establish security arguments for signature schemes no longer work since, during the collusion between the signer, the attacker and the random oracle, we lose control over the value that the signer receives: it no longer comes from the random oracle, but from the attacker. As a consequence, the signer cannot be simulated without the secret key, otherwise the signature scheme would be universally forgeable.

In order to overcome this problem, [25, 27, 28] use the concept of the “witness indistinguishable” proofs. This notion was defined by Feige and Shamir in [14] for the purpose of identification. In such a proof system:

- Many secret keys are associated to a same public key.
- The views of two proofs using two distinct secret keys (witnesses) associated to a same public key are indistinguishable, even from the point of view of the verifier.
- The knowledge of two distinct secret keys associated to a same public one provides the solution of a difficult problem.

For example, in the Fiat-Shamir protocol [17], the verifier cannot distinguish which square root the prover uses, and with probability $\frac{1}{2}$, two distinct square roots provide the factorization of the modulus. Okamoto, in [23], proposed a witness indistinguishable adaptation of both the Schnorr [31] and the Guillou-Quisquater [20] identification schemes.

As was already remarked, the technical difficulty to be overcome comes from the fact that, in the colluding step, one can no longer simulate the signer without the secret key. One thus uses a scheme which admits more than one secret key for a given public key. This makes the collusion possible and one constrains the attacker to output a different secret key.

Let us just provide the simplest candidate scheme, derived from the Okamoto's adaptation of the Schnorr's scheme [31, 32, 23].

The Okamoto-Schnorr Blind Signature Scheme The scheme uses two large primes p and q such that $q \mid (p - 1)$, and two elements $g, h \in \mathbb{Z}_p^*$ of order q . The authority chooses a secret key $(r, s) \in (\mathbb{Z}_q^*)^2$ and publishes the public key, $y = g^{-r}h^{-s} \bmod p$. We assume that the function f outputs elements in $\mathbb{Z}q$ and that $\lceil \log q \rceil = k$, where k is, as usual, the security parameter. The protocol by which the user obtains a blind signature of the message m is as follows:

- The authority chooses $(t, u) \in (\mathbb{Z}_t^*)^2$, computes and sends the commitment $a = g^t h^u \bmod p$.
- The user chooses $\beta, \gamma, \delta \in \mathbb{Z}q$ and blinds a into $\alpha = ag^\beta h^\gamma y^\delta \bmod p$. He computes the challenge $\varepsilon = f(m, \alpha)$ and sends $e = \varepsilon - \delta \bmod q$ to the authority.
- The authority computes $R = t + er \bmod q$ and $S = u + es \bmod q$, and sends the pair (R, S) which satisfies $a = g^R h^S y^e \bmod p$;
- the user computes $\rho = R + \beta \bmod q$ and $\sigma = S + \gamma \bmod q$.

Straightforward computations show that $\alpha = g^\rho h^\sigma y^\varepsilon \bmod p$, with $\varepsilon = f(m, \alpha)$. Security arguments follow from the theorem below.

Theorem 2.4 Consider the Okamoto-Schnorr blind signature scheme in the random oracle model. Let \mathcal{A} be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by Q and ℓ the number of queries that \mathcal{A} can ask to the random oracle and the number of queries that \mathcal{A} can ask to the authority. Assume that, with a time bound T , \mathcal{A} performs, with probability $\varepsilon \geq 4Q^{\ell+1}/q$, an $(\ell, \ell + 1)$ -forgery. Then there is another machine which has control over \mathcal{A} and solves the discrete logarithm of h relative to g in expected time $T' \leq 10^6(\ell + 1)^2 k^2 QT/\varepsilon$,

The technique can be easily applied to all the other schemes that come from witness indistinguishable protocols. Especially, the Okamoto version of the Guillou-Quisquater identification scheme provides a provably secure blind signature scheme relative to the security of RSA. Furthermore, [27] proposes blind signature schemes derived from the Fiat-Shamir identification scheme [17] and from the Ong-Schnorr identification scheme [24], which are clearly witness indistinguishable. The resulting schemes admit security arguments relative to factorization.

However, these security results left an open problem: the complexity of the reduction is polynomial in the size of the key but not in ℓ . The theorem thus only provides security arguments against strong “one-more” forgeries. In fact, the reduction requires $\varepsilon \geq 4Q^{\ell+1}/q$, which implies a polylogarithmically bounded number of interactions with the authority. Schnorr [33]

thereafter proved this is actually optimal: interactions with the signer provide the adversary with a system of equations which solution leads to a forgery. With more than polylogarithmically many interactions, this system does not have any solution. Above this limit, the system may have solutions. Schnorr thus introduced the new ROS-problem (find an Overdetermined, Solvable system of linear equations modulo q , with Random inhomogenities), which consists in solving this system. Under the intractability assumption of this problem, a higher bound can be proven.

On a more theoretical level, Juels et al. [21] suggested a provably secure construction using the provably secure signature scheme of Naor and Yung [22] and the Two-Party Completeness Theorem [19]. Nevertheless, their construction is theoretical.

About the most efficient construction, proposed by Chaum and presented above, under the *RSA blind signature*, [4, 5] proposed a proof of security under a new assumption, the so-called *one-more inversion RSA*.

2.6.4 Fair Blind Signatures

After another proposal [3] secure after only polylogarithmically many interactions, Abe [1] presented the first efficient scheme secure after polynomially many interactions, under the decisional Diffie-Hellman problem. Actually, this scheme is no longer a truly “blind” signature scheme, since the blindness is not unconditional, but relies on the decisional Diffie-Hellman problem. On the other hand, this helped them to design a fair blind signature scheme [2], where anonymity can be revoked by an authority.

Acknowledgments

Contributors from ECRYPT: David Pointcheval.

References

- [1] M. Abe. A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures. In *Eurocrypt '01*, LNCS 2045, pages 136–151. Springer-Verlag, Berlin, 2001.
- [2] M. Abe and M. Ohkubo. Provably Secure Fair Blind Signatures with Tight Revocation. In *Asiacrypt '01*, LNCS 2248. Springer-Verlag, Berlin, 2001.
- [3] M. Abe and T. Okamoto. Provably Secure Partially Blind Signature. In *Crypto '00*, LNCS 1880, pages 271–286. Springer-Verlag, Berlin, 2000.
- [4] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The Power of RSA Inversion Oracles and the Security of Chaum’s RSA Blind Signature Scheme. In *Financial Cryptography '01*, LNCS 2339. Springer-Verlag, Berlin, 2001.
- [5] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The One-More-RSA-Inversion Problems and the Security of Chaum’s Blind Signature Scheme. *Journal of Cryptology*, 16(3):185–215, 2003.

- [6] S. A. Brands. An Efficient Off-Line Electronic Cash System Based on the Representation Problem. Technical Report CS-R9323, CWI, Amsterdam, 1993.
- [7] S. A. Brands. Untraceable Off-Line Cash in Wallets with Observers. In *Crypto '93*, LNCS 773, pages 302–318. Springer-Verlag, Berlin, 1994.
- [8] S. A. Brands. A Note on Parallel Executions of Restrictive Blind Issuing Protocols for Secret-Key Certificates. Technical Report CS-R9519, CWI, Amsterdam, 1995.
- [9] S. A. Brands. More on Restrictive Blind Issuing of Secret-Key Certificates in Parallel Mode. Technical Report CS-R9534, CWI, Amsterdam, 1995.
- [10] S. A. Brands. Off-Line Electronic Cash Based on Secret-Key Certificates. In *LATIN '95*, LNCS 911, pages 131–166. Springer-Verlag, Berlin, 1995.
- [11] S. A. Brands. Restrictive Blind Issuing of Secret-Key Certificates in Parallel Mode. Technical Report CS-R9523, CWI, Amsterdam, 1995.
- [12] D. Chaum. Blind Signatures for Untraceable Payments. In *Crypto '82*, pages 199–203. Plenum, New York, 1983.
- [13] D. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In *Crypto '88*, LNCS 403, pages 319–327. Springer-Verlag, Berlin, 1989.
- [14] U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *Proc. of the 22nd STOC*, pages 416–426. ACM Press, New York, 1990.
- [15] N. Ferguson. Extensions of Single Term Coins. In *Crypto '93*, LNCS 773, pages 292–301. Springer-Verlag, Berlin, 1994.
- [16] N. Ferguson. Single Term Off-Line Coins. In *Eurocrypt '93, Berlin*, LNCS 765, pages 318–328. Springer-Verlag, 1994.
- [17] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions of Identification and Signature Problems. In *Crypto '86*, LNCS 263, pages 186–194. Springer-Verlag, Berlin, 1987.
- [18] M. Franklin and M. Yung. Secure and Efficient Off-Line Digital Money. In *Proc. of the 20th ICALP*, LNCS 700, pages 265–276. Springer-Verlag, Berlin, 1993.
- [19] O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority. In *Proc. of the 19th STOC*, pages 218–229. ACM Press, New York, 1987.
- [20] L. C. Guillou and J.-J. Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. In *Eurocrypt '88*, LNCS 330, pages 123–128. Springer-Verlag, Berlin, 1988.
- [21] A. Juels, M. Luby, and R. Ostrovsky. Security of Blind Digital Signatures. In *Crypto '97*, LNCS 1294, pages 150–164. Springer-Verlag, Berlin, 1997.
- [22] M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. In *Proc. of the 21st STOC*, pages 33–43. ACM Press, New York, 1989.

- [23] T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In *Crypto '92*, LNCS 740, pages 31–53. Springer-Verlag, Berlin, 1992.
- [24] H. Ong and C.P. Schnorr. Fast Signature Generation with a Fiat-Shamir-Like Scheme. In *Eurocrypt '90*, LNCS 473, pages 432–440. Springer-Verlag, Berlin, 1991.
- [25] D. Pointcheval and J. Stern. Provably Secure Blind Signature Schemes. In *Asiacrypt '96*, LNCS 1163, pages 252–265. Springer-Verlag, Berlin, 1996.
- [26] D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *Eurocrypt '96*, LNCS 1070, pages 387–398. Springer-Verlag, Berlin, 1996.
- [27] D. Pointcheval and J. Stern. New Blind Signatures Equivalent to Factorization. In *Proc. of the 4th CCS*, pages 92–99. ACM Press, New York, 1997.
- [28] D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
- [29] C. Radu, R. Govaerts, and J. Vanderwalle. A Restrictive Blind Signature Scheme with Applications to Electronic Cash. In *Communications and Multimedia Security II*, pages 196–207. Chapman & Hall, London, 1996.
- [30] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
- [31] C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Crypto '89*, LNCS 435, pages 235–251. Springer-Verlag, Berlin, 1990.
- [32] C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
- [33] C. P. Schnorr. Security of Blind Discrete Log Signatures against Interactive Attacks. In *ICICS '01*, LNCS, pages 1–12. Springer-Verlag, Berlin, 2001.
- [34] B. Schoenmakers. An Efficient Electronic Payment System withstanding Parallel Attacks. Technical Report CS-R9522, CWI, Amsterdam, 1995.

2.7 Undeniable signatures

2.7.1 Introduction

Digital signatures are the analogue of handwritten signatures, and as such, they must capture their properties, namely authentication, integrity and non-repudiation. The important fact is that anybody having a signature can be convinced of all these properties if it is a real one. However, unlike handwritten signatures, digital signatures can be “copy-cloned”, and therefore, authenticated documents can be easily disseminated. Therefore, in many electronic applications, it is necessary that the verification of a signature be controlled by the signer or a third party.

This is the main motivation of the introduction, in 1989 at Crypto [7], by Chaum and van Antwerpen of the so-called *undeniable signatures*. The underlying idea is that someone who

wants to verify a signature has to interact with the signer (a better name might have been *undeniable signatures*). Here is a definition of undeniable signature.

2.7.2 Definition

Definition 2.5 [Undeniable Signature] Given an integer k , an *undeniable signature scheme* US with security parameter k is defined by the following:

- a *common parameter generation algorithm* US.Setup;
- a *key generation algorithm* US.KeyGen;
- a *signing algorithm* US.Sign;
- *confirming/denying protocols* US.{Confirm, Deny};

and should satisfy the following (informally stated) security properties:

1. *completeness and soundness* the confirmation protocol works on valid signatures and it is computationally infeasible for anyone other than the original signer to confirm a signature;
2. *undeniable* it is computationally infeasible for a signer to successfully run a denial protocol on a valid signature which they have generated;
3. *unforgeability* it should be computational infeasible for an adversary to create a valid undeniable signature;
4. *invisibility*: given a valid undeniable signature σ , it is computationally infeasible without the knowledge of the signing key or without performing confirm/deny protocols with the signer, to determine which message m has signature σ ;
5. *anonymity*: given a message m and a valid undeniable signature σ on m , it is computationally infeasible without the knowledge of the signers' keys or without performing confirm/deny protocols with various signers, to determine which signer has signed the message.
6. *non-transferability*: a verifier who has participated in an execution of the confirming/denying protocols with a signer cannot prove to a third party the validity/invalidity of a signature.

When they introduced this concept, Chaum and van Antwerpen had in mind the problem of software vendors who want to control the distribution of the licence. These signatures have then lots of applications. In practice, the interactive protocol is an interactive zero-knowledge proof, or, to face some attacks like blackmailing [15], an interactive designated verifier proof [16].

The security requirements for undeniable signatures were gradually developed over time. The idea of invisibility was introduced by Chaum, van Heijst and Pfitzmann [8]. In [8] invisibility

was defined in terms of simulatability. Camenisch and Michels [4] phrased the notion in terms of distinguishing whether a (valid) signature σ corresponds to a message m_0 or m_1 . The notion of anonymity was proposed by Galbraith and Mao [11]. They show that anonymity and invisibility are equivalent notions for schemes with certain additional properties.

2.7.3 Convertible and delegated signatures

The concept has then been refined to obtain *convertible undeniable signatures* introduced by Boyar, Chaum, Damgård and Pedersen at Crypto'90 [3]: it is sometimes important that an undeniable signature becomes universally verifiable. The conversion can be individual (for one signature) or universal (for all signatures, past and future).

Another variant of the concept is what one could call *delegated signatures*. In this case, not only the signer can convince a verifier, but also third party. *Directed signatures* introduced in 1993 by Lim and Lee [23, 24], *nominative signatures* [17], *(designated) confirmer signatures* [6], or *limited verifier signatures* [1] [9] are among the best known examples. They are still undeniable signatures, and the definition, with the corresponding security requirements can easily be deduced from Def. 2.5.

2.7.4 Proposed schemes

The original undeniable signature scheme (based on discrete logarithms in finite fields) is given in [7] and an improved denial method is given in [5]. The security of the original scheme has recently been analysed. Okamoto and Pointcheval [29] studied the security against forgery under adaptive attacks, and this led to the introduction of the notion of gap problems. In the extended version of [11] (see [12]) Galbraith and Mao show that a natural variant of the [7] scheme has invisibility and anonymity.

Some generic results have been given. For example, Okamoto [28] relates designated confirmer signatures and public key encryption. Michels and Stadler [26] give generic constructions for confirmer signatures. Note that these constructions date from earlier than the security definitions of [4] and [11] and so the proposals do not necessarily guarantee full security.

Many schemes have then been proposed, based upon classical signatures, such as Schnorr [25], El Gamal [10] and RSA [14, 13, 11]. Very recently, Monnerat and Vaudenay [27] proposed short undeniable signatures based on the computation of characters which do not provide the conversion property. In [21], Laguillaumie and Vergnaud formalized a new notion of conversion, the *time-selective conversion*, and proposed a pairing-based scheme with this property, *i.e.* signatures pertaining to a specific time period can be individually converted. An identity-based variant was also proposed by Libert and Quisquater in [22].

2.7.5 Non Transferable signatures

Designated verifier signatures (DVS) comes from designated verifier proofs introduced at Eurocrypt'96 by Jakobsson, Sako and Impagliazzo [16], to fix some problems of undeniable signatures. To face blackmailing attacks, for instance, designated verifier proofs are preferred

to zero-knowledge proofs during the interactive protocols. Jakobsson *et al.* obtained designated verifier signature from the proof via the Fiat-Shamir heuristic. DVS are intended to a specific and unique designated verifier, who is the only one able to check their validity. This verifier cannot convince another party that the signature is actually valid, essentially because he can also perform this signature by himself. The anonymity properties required can be found in [19].

In [30], Rivest, Shamir and Tauman introduced the notion of ring signatures. By setting the size of the ring to two members, these signatures also provide DVS. Recently, in [31], Saeednia, Kremer and Markowitch proposed very efficient DVS with signatures *à la* Schnorr. In [34], Susilo, Zhang and Mu proposed an identity-based strong DVS which is a pairing-based variant of [31] and whose security is investigated in the same model. In [32], Steinfeld, Bull, Wang and Pieprzyk proposed a formalization of *Universal DVS* (UDVS). These are ordinary digital signatures with the additional functionality that any holder of a signature is able to convert it into a DVS specified to any designated verifier of his choice. At PKC'04 [33], Steinfeld, Wang and Pieprzyk proposed three new DVS constructions based on Schnorr and RSA signatures. A generalization of the concept is described in [20]. It exists also identity based variant [34]. They find many applications in electronic voting, or contract signing.

A similar concept is the chameleon signatures. Basically, these are signatures for which the underlying hash function is a *chameleon hash function*, which is essentially a trapdoor collision-resistant hash function. They have been introduced by Krawczyk and Rabin in [18], and roughly speaking, allow the trapdoor owner to produce a collision. The chameleon signatures are designated verifier, but they are non-repudiable, in case of dispute. The related problem of key exposure is investigated in [2].

2.7.6 Open Problems

We can imagine lots of variants of these signatures to fit in real applications. The security of all these signatures is usually investigated in the random oracle model, and it would be interesting to have proofs in the standard model.

Acknowledgments

Contributors from ECRYPT: Steven Galbraith, Fabien Laguillaumie.

References

- [1] S. Araki, S. Uehara, K. Imamura: The Limited Verifier Signature and Its Application. IEICE Trans. Fundamentals, Vol. E82-A (1), 63–68 (1999)
- [2] G. Ateniese, B. de Medeiros: On the Key Exposure Problem in Chameleon Hashes. Proc. SCN'04, Springer LNCS, to appear.
- [3] J. Boyar, D. Chaum, I. B. Damgård, T.P. Pedersen: Convertible undeniable signatures. Proc. of Crypto'90, Springer LNCS Vol. 537, 189–205 (1991)

- [4] J. Camenisch and M. Michels, Confirmer signature schemes secure against adaptive adversaries, in B. Preneel (ed.), EUROCRYPT 2000, Springer LNCS 1870 (2000) 243–258.
- [5] Chaum, D. newblock Zero-knowledge undeniable signatures, in I.B. Damgård (ed.), CRYPTO '90, Springer LNCS 473 (1991) 458–464.
- [6] D. Chaum: Designated Confirmer Signatures. Proc. of Eurocrypt'94, Springer LNCS Vol. 950, 86–91 (1995)
- [7] D. Chaum, H. van Antwerpen: Undeniable Signatures. Proc. of Crypto'89, Springer LNCS Vol. 435, 212–216 (1989)
- [8] D. Chaum, E. van Heijst and B. Pfitzmann, Cryptographically strong undeniable signatures, unconditionally secure for the signer, in J. Feigenbaum (ed.), CRYPTO '91, Springer LNCS 576 (1992) 470–484.
- [9] X. Chen, F. Zhang, K. Kim: Limited Verifier Signature from Bilinear Pairings. Proc. of ACNS'04, Springer LNCS Vol. 3089, 135–148 (2004)
- [10] I. Damgard, T.P. Pedersen: New convertible undeniable signature schemes. Proc. of Eurocrypt'96, Springer LNCS Vol. 1070, 372–386 (1996)
- [11] S. Galbraith, W. Mao: Invisibility and anonymity of undeniable and confirmer signatures. Proc. of CT-RSA 2003, Springer LNCS Vol. 2612 80–97 (2003)
- [12] S. Galbraith, W. Mao: Invisibility and anonymity of undeniable and confirmer signatures (full version). <http://www.isg.rhul.ac.uk/~sdg/>
- [13] S. Galbraith, W. Mao, K.G. Paterson: RSA-based undeniable signatures for general moduli. Proc. of CT-RSA 2002, Springer LNCS Vol. 2271, 200–217 (2002)
- [14] R. Gennaro, H. Krawczyk, T. Rabin: RSA-based undeniable signatures. Proc. of Crypto'97, Springer LNCS Vol. 1294, 132–149 (1997)
- [15] M. Jakobsson: Blackmailing using undeniable signatures. Proc. of Eurocrypt'94, Springer LNCS Vol. 950, 425–427 (1994)
- [16] M. Jakobsson, K. Sako, R. Impagliazzo: Designated Verifier Proofs and their Applications. Proc. of Eurocrypt'96, Springer LNCS Vol. 1070, 142–154 (1996)
- [17] S. J. Kim, S. J. Park, D. H. Won: A Nominative Signature. Proc. of CISC'94, Conference on Information Security and Cryptology, Vol. 4 (1) (1994)
- [18] H. Krawczyk, T. Rabin: Chameleon Signatures. Proc. of NDSS 2000, 143–154 (2000)
- [19] F. Laguillaumie, D. Vergnaud: Designated Verifier Signatures: Anonymity and Efficient Construction from *any* Bilinear Map. Proc. of SCN'04, Springer LNCS, to appear.
- [20] F. Laguillaumie, D. Vergnaud: Multi-Designated Verifiers Signatures. Proc. of ICICS'04, Springer LNCS, to appear.
- [21] F. Laguillaumie, D. Vergnaud: Time-Selective Convertible Undeniable Signatures. Proc. of CT-RSA 2005, Springer LNCS, to appear.

- [22] B. Libert, J.-J. Quisquater: Identity Based Undeniable Signatures. Proc. of CT-RSA 2004, Springer LNCS Vol. 2964, 112–125 (2004)
- [23] C. H. Lim and P. J. Lee: Modified Maurer-Yacobi’s Scheme and its Applications. Proc. of Auscrypt’92, Springer LNCS Vol. 718, 308–323 (1993)
- [24] C. H. Lim and P. J. Lee: Directed Signatures and Application to Threshold Cryptosystems, Security Protocols, Springer LNCS Vol. 1189, 131–138 (1996)
- [25] M. Michels, M. Stadler: Efficient Convertible Undeniable Signature Schemes. Proc. of SAC’97, 231–244 (1997)
- [26] M. Michels and M. Stadler, Generic constructions for secure and efficient confirmer signature schemes, in K. Nyberg (ed.) EUROCRYPT ’98, Springer LNCS 1403 (1998) 406–421.
- [27] J. Monnerat, S. Vaudenay: Undeniable Signatures Based on Characters: How to Sign with One Bit. Proc. of PKC 2004, Springer LNCS Vol. 2947, 69–85 (2004)
- [28] T. Okamoto, Designated confirmer signatures and public key encryption are equivalent, in Y. G. Desmedt (ed.), CRYPTO ’94, Springer LNCS 839 (1994) 61–74.
- [29] T. Okamoto and D. Pointcheval, The gap problems: a new class of problems for the security of cryptographic schemes, in K. Kim (ed.) PKC 2001, Springer LNCS 1992 (2001) 104–118.
- [30] R. L. Rivest, A. Shamir, Y. Tauman: How to Leak a Secret. Proc. of Asiacrypt’01, Springer LNCS Vol. 2248, 552–565 (2001)
- [31] S. Saeednia, S. Kremer, O. Markowitch: An Efficient Strong Designated Verifier Signature Scheme. Proc. of ICISC 2003, Springer LNCS Vol. 2836, 40–54 (2003)
- [32] R. Steinfeld, L. Bull, H. Wang, J. Pieprzyk: Universal Designated Verifier Signatures. Proc. of Asiacrypt’03, Springer LNCS Vol. 2894, 523–542 (2003)
- [33] R. Steinfeld, H. Wang, J. Pieprzyk: Efficient Extension of Standard Schnorr/RSA signatures into Universal Designated-Verifier Signatures. Proc. of PKC’04, Springer LNCS Vol. 2947, 86–100 (2004)
- [34] W. Susilo, F. Zhang, Y. Mu: Identity-based Strong Designated Verifier Signatures Schemes. Proc. of ACISP’04, Springer LNCS Vol. 3108, 313–324 (2004)

2.8 Group/ring signatures

2.8.1 Introduction

In [CvH91] Chaum and van Heyst introduced the notion of a *group signature* scheme. Informally, in a group signature scheme, any member of the group can sign a document and any verifier can confirm that the signature has been computed by a member of the group. Moreover, group signatures are anonymous and unlinkable for every verifier except, in case of a dispute, for a given authority that knows some special information.

The notion of a *ring signature* scheme has been formalized in [30]. Here an entity can sign a message on behalf on a group that includes himself. Again a verifier is convinced that the signature has been computed by a member of the group but the identity of the signer is not disclosed. A crucial property of a ring signature scheme is that there is no special entity (e.g., no group manager).

Motivation. A straightforward application of group signatures is the possibility of a company of authenticating some data (e.g., a document or a contract proposal) that have to be publicly verifiable. In this case any member of the company can compute a signature for such data, everybody can verify the authenticity of the data and a special authority inside the company can discover the identity of the signer.

Ring signatures are preferred in a dynamic setting where a trusted third party is not available and in general, where there is no coordination/cooperation between the users. More specifically, a ring signature scheme can be used by only requiring that users have public keys. Instead, a group signature requires a strong set-up assumption that involve the generation of public parameters and procedures for making the group. But the drawback of ring signatures is that their size are proportional to the number of group members, which is not the case for efficient group signature schemes.

2.8.2 Description of the problem

Group signatures. A group signature scheme is a quintuple of algorithms `SETUP`, `JOIN`, `SIGN`, `VERIFY`, `OPEN`² such that:

- `SETUP`: on input a security parameter, the group public key and the group secret key for the group manager are generated;
- `JOIN`: a user becomes a new group member by interacting with the group manager. The output of the interaction is a membership certificate and a membership secret key for the new user;
- `SIGN`: on input a group public key, a membership certificate, a membership secret, and a message outputs a group signature;
- `VERIFY`: verifies the correctness of a signature with respect to a message and a group public key;
- `OPEN`: on input a message, a corresponding signature, a group public key and a group manager's secret key discovers the identity of the signer.

Two main security models have been presented and we refer to them as the *weak* one and *strong* one respectively. Informally a weak-secure group signature scheme [ACJT00] enjoys the following properties:

- correctness: a correct signature is successfully verified;

²In [BMW03] since the size of the group is static, `SETUP` and `JOIN` are merged in the same algorithm.

- unforgeability: only group members can compute signatures;
- anonymity: a signer is anonymous unless the group secret key is known;
- unlinkability: different signatures cannot be linked to the same signer;
- exculpability: nobody can sign on behalf of a given member;
- traceability: the group manager can discover the identity of the signer of a message; in case a colluding subset of group members generates a signature then the group manager discovers the identity of at least one of the coalition.
- framing: the group manager cannot falsely accuse a group member of having produced a particular group signature.

In [BMW03] the notion of strong-secure group signature scheme has been proposed by only considering the two following properties: *full-anonymity* where the adversary tries to distinguish a signature computed by one of two possible members and *full-traceability* where a colluding group cannot compute an untraceable signature even in case the members possess the secret key of the group manager. This two properties imply all the properties listed above for the weak notion of security (except framing).

Ring signatures. A ring signature scheme uses a classical signature scheme (e.g., RSA or Rabin) and every potential group member owns a pair of public and secret signature keys of this classical scheme. A ring signature scheme then consists of two procedures. The former, referred to as **Ring-Sign**, is used to compute a signature on input a secret key, the public keys of the other members of the group and a message. The latter, referred to as **Ring-Verify** is used to verify a signature on input a message, the signature and the public keys of the members of the group.

There are two security requirements with respect to ring signatures. First of all, it should be hard for an entity U to sign a message with respect to a group that does not include U . Secondly, a verifier cannot link different ring signatures of the same member and the probability that the identity of the signer is disclosed in a group of size k is at most $1/k$.

2.8.3 State of the art

Group signatures

Efficient weak-secure constructions. In [ACJT00] a weak-secure group signature scheme is presented in the random oracle model. The system is efficient and provable weak-secure under the DDH and strong RSA assumptions. A similar result has been recently shown in [DAK04].

General strong-secure constructions. In [BMW03] a strong-secure group signature scheme is presented in the standard model. The system is based on general complexity-based assumptions, each tool can be implemented assuming existence of one-way trapdoor

permutations. The resulting scheme is not efficient and is actually a *relaxed* group signature scheme as an additional trusted third party that knows users secret keys is required. The case of dynamic groups is discussed in [BSZ05].

Efficient strong-secure constructions. Efficient strong-secure constructions have been recently proposed in [KY04, CL04, CG04, BBS04, NSN04] in the random oracle model. In some cases, strong security is actually achieved by using some variations of the security model presented in [BMW03] that however do not resort to relaxed group signature schemes.

Revocation. Member revocation is one of the major weaknesses for all known group signature schemes. The use of *accumulators* [CL02] allows many group signature schemes to have an efficient revocation that however requires a local computation for signers and verifiers for each member of the group that is added or revoked. Recently, in [BS04], an efficient linkable group signature scheme is presented where such a computation is only required by verifiers. But, in case the group signatures must not be linkable, this solution is not really practical.

Ring signatures In [30] an efficient ring signature scheme is presented. The scheme is based on a joint use of a trapdoor permutation and of a symmetric encryption scheme. When $\text{RSA}(3, n)$ is used as trapdoor permutation, the scheme requires 1 modular exponentiation and for each member of the group 2 modular multiplications and 1 symmetric encryption. If instead Rabin's cryptosystem is used as trapdoor permutation, the performance of the scheme is improved by 1 modular multiplication per user. The scheme is proven to be secure in the *ideal cipher* model, where it is assumed the existence of a family of keyed random permutations that can be oracle accessed.

The scheme has been later improved in [BSS02] where an efficient ring signature scheme is presented in the random oracle model.

Recently, the first ring signature scheme with constant-size signatures has been presented in [DAK04]. The scheme is efficient and proven secure under the strong RSA assumption, in the random oracle model but it has the drawback of being constant-size only for static groups.

2.8.4 Open Problems

A first interesting open problem is the existence of a group signature scheme under weaker complexity-theoretic assumptions. A second interesting problem is the existence of an efficient group signature (resp., ring signature) scheme that does not use random oracles. Then, it can be useful to find constant-size ring signatures for dynamic groups. Finally, an open problem is the existence of more practical solutions for member revocations in group signatures.

Acknowledgments

Contributors from ECRYPT: Sébastien Canard, Ivan Visconti.

References

- [ACJT00] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *Advances in Cryptology - Crypto '00*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer-Verlag, 2000.
- [BBS04] D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. In *Advances in Cryptology - Crypto '04*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer-Verlag, 2004.
- [BMW03] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *Advances in Cryptology - Eurocrypt '03*, volume 2045 of *Lecture Notes in Computer Science*, pages 614–629. Springer-Verlag, 2003.
- [BS04] D. Boneh and H. Shacham. Group Signatures with Verifier-Local Revocation. In *In proceedings of the 11'th ACM conference on Computer and Communications Security (CCS 04)*, 2004.
- [BSS02] E. Bresson, J. Stern, and M. Szydlo. Threshold Ring Signatures and Applications to Ad-hoc Groups. In *Advances in Cryptology - Crypto '02*, volume 2442 of *Lecture Notes in Computer Science*, pages 465–480. Springer-Verlag, 2002.
- [BSZ05] M. Bellare, H. Shi, and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In *CT-RSA '05*, *Lecture Notes in Computer Science*. Springer-Verlag, 2005.
- [CG04] J. Camenisch and J. Groth. Group Signatures: Better Efficiency and New Theoretical Aspects. In *In Forth Conference on Security in Communication Networks - SCN '04*, *Lecture Notes in Computer Science*. Springer-Verlag, 2004.
- [CL02] J. Camenisch and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Advances in Cryptology - Crypto '02*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer-Verlag, 2002.
- [CL04] J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *Advances in Cryptology - Crypto '04*, volume 3152 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.
- [CvH91] D. Chaum and E. van Heyst. Group Signatures. In D. W. Davies, editor, *Advances in Cryptology - Eurocrypt '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer-Verlag, 1991.
- [DAK04] Y. Dodis and V. Shoup. Anonymous Identification in Ad Hoc Groups. In *Advances in Cryptology - Eurocrypt '04*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer-Verlag, 2004.

- [KY04] A. Kiayias and M. Yung. Group Signatures: Provable Security, Efficient Constructions and Anonymity from Trapdoor-Holders. In *e-print archive*, available at: <http://eprint.iacr.org/2004/076/>, 2004.
- [NSN04] L. Nguyen and R. Safavi-Naini. Efficient and Provably Secure Trapdoor-free Group Signature Schemes from Bilinear Pairings. In *Asiacrypt '04*, Lecture Notes in Computer Science. Springer-Verlag, 2004.
- [RST01] R. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. In C. Boyd, editor, *Advances in Cryptology – Asiacrypt '01*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer-Verlag, 2001.

3 Encryption with special properties

3.1 Searchable encryption

3.1.1 Description and Motivations

Suppose user Alice wishes to read her email on a number of devices: laptop, desktop, pager, etc. Alice’s mail gateway is supposed to route email to the appropriate device based on the keywords in the email. For example, when Bob sends email with the keyword “urgent” the mail is routed to Alice’s pager. When Bob sends email with the keyword “lunch” the mail is routed to Alice’s desktop for reading later. One expects each email to contain a small number of keywords. For example, all words on the subject line as well as the sender’s email address could be used as keywords. The mobile people project [MRS⁺99] provides this email processing capability.

Now, suppose Bob sends encrypted email to Alice. As usual, Bob encrypts the email and all keywords using Alice’s public key. In this case the mail gateway cannot see the keywords and hence cannot make routing decisions. As a result, the mobile people project is unable to process secure email without violating user privacy. Our goal is to enable the gateway to test whether “urgent” is a keyword in the email, but the gateway should learn nothing else about the email. More generally, Alice should be able to specify a few keywords that the mail gateway can search for, but learn nothing else about incoming mail.

To do so, Bob encrypts his email using a standard public key system. He then appends to the resulting ciphertext a *searchable public-key encryption* (SES) of each keyword. To send a message M with keywords W_1, \dots, W_m Bob sends

$$E_{A_{pub}}(M) \parallel \text{SES}(A_{pub}, W_1) \parallel \dots \parallel \text{SES}(A_{pub}, W_m)$$

where A_{pub} is Alice’s public key. The point of searchable encryption is that Alice can give the gateway a certain trapdoor T_W that enables the gateway to test whether one of the keywords associated with the message is equal to the word W of Alice’s choice. Given $\text{SES}(A_{pub}, W')$ and T_W the gateway can test whether $W = W'$. If $W \neq W'$ the gateway learns nothing more about W' . Note that Alice and Bob do not communicate in this entire process. Bob generates the searchable encryption for W' just given Alice’s public key.

In some cases, it is instructive to view the email gateway as an IMAP or POP email server. The server stores many emails and each email contains a small number of keywords. As before, all these emails are created by various people sending mail to Alice encrypted using her public key. We want to enable Alice to ask queries of the form: do any of the messages on the server contain the keyword “urgent”? Alice would do this by giving the server a trapdoor T_W , thus enabling the server to retrieve emails containing the keyword W . The server learns nothing else about the emails.

The concept of a searchable encryption was first introduced in the context of a *operator oriented encryption* by [Des93]. Roughly speaking, in an operator oriented encryption, we would have that, given an encrypted text corresponding with a plaintext and a signed operator it would be easy to compute $\text{Operator}(\text{Plaintext})$ without revealing anything additionally about the rest of the plaintext.

Formal definition We review the formal definition of a searchable encryption scheme of [BDCOP04].

Definition 3.1 A searchable encryption scheme (SES) consists of the following polynomial time randomized algorithms:

1. $\text{KeyGen}(s)$: Takes a security parameter, s , and generates a public/private key pair A_{pub}, A_{priv} .
2. $\text{SES}(A_{pub}, W)$: for a public key A_{pub} and a word W , produces a searchable ciphertext of W .
3. $\text{Trapdoor}(A_{priv}, W)$: given Alice’s private key and a word W produces a trapdoor T_W .
4. $\text{Test}(A_{pub}, S, T_W)$: given Alice’s public key, a searchable encryption $S = \text{SES}(A_{pub}, W')$, and a trapdoor $T_W = \text{Trapdoor}(A_{priv}, W)$, outputs ‘yes’ if $W = W'$ and ‘no’ otherwise.

Alice runs the KeyGen algorithm to generate her public/private key pair. She uses Trapdoor to generate trapdoors T_W for any keyword W that she wants the mail server or mail gateway to search for. The mail server uses the given trapdoors as input to the $\text{Test}()$ algorithm to determine whether a given email contains one of the keywords W specified by Alice.

Next, we define security for an SES. We need to ensure that an $\text{SES}(A_{pub}, W)$ does not reveal any information about W , unless T_W is available. We define security against an active attacker who is able to obtain trapdoors T_W for any W of his choice. Even under such an attack the attacker should not be able to search for a keyword W' for which he did not obtain the trapdoor. Formally, we define security against an active attacker \mathcal{A} using the following game between a challenger and the attacker (the security parameter s is given to both players as input).

SES Security game:

1. The challenger runs the $\text{KeyGen}(s)$ algorithm to generate A_{pub} and A_{priv} . It gives A_{pub} to the attacker.
2. The attacker can adaptively ask the challenger for the trapdoor T_W for any keyword $W \in \{0, 1\}^*$ of his choice.

3. At some point, the attacker \mathcal{A} sends the challenger two words W_0, W_1 on which it wishes to be challenged. The only restriction is that the attacker did not previously ask for the trapdoors T_{W_0} or T_{W_1} . The challenger picks a random $b \in \{0, 1\}$ and gives the attacker $C = \text{SES}(A_{pub}, W_b)$. We refer to C as the challenge SES.
4. The attacker can continue to ask for trapdoors T_W for any keyword W of his choice as long as $W \neq W_0, W_1$.
5. Eventually, the attacker \mathcal{A} outputs $b' \in \{0, 1\}$ and wins the game if $b = b'$.

In other words, the attacker wins the game if he can correctly guess whether he was given the SPKE for W_0 or W_1 . We define \mathcal{A} 's advantage in breaking the SES as

$$\text{Adv}_{\mathcal{A}}(s) = \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

Definition 3.2 We say that searchable encryption scheme SES is semantically secure against an adaptive chosen keyword attack if for any polynomial-time attacker \mathcal{A} we have that $\text{Adv}_{\mathcal{A}}(s)$ is a negligible function.

3.1.2 State of the art

In this section we review the main constructions known for searchable encryptions.

A SES using bilinear maps The construction discussed in this section appears in [BDCOP04]. They use two groups G_1, G_2 of prime order p and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ between them. The map satisfies the following properties:

1. Computable: given $g, h \in G_1$ there is a polynomial time algorithms to compute $e(g, h) \in G_2$.
2. Bilinear: for any integers $x, y \in [1, p]$ we have $e(g^x, g^y) = e(g, g)^{xy}$
3. Non-degenerate: if g is a generator of G_1 then $e(g, g)$ is a generator of G_2 .

The size of G_1, G_2 is determined by the security parameter.

The construction also needs hash functions $H_1 : \{0, 1\}^* \rightarrow G_1$ and $H_2 : G_2 \rightarrow \{0, 1\}^{\log p}$. The SES of [BDCOP04] works as follows:

- **KeyGen**: The input security parameter determines the size, p , of the groups G_1 and G_2 . The algorithm picks a random $\alpha \in Z_p^*$ and a generator g of G_1 . It outputs $A_{pub} = [g, h = g^\alpha]$ and $A_{priv} = \alpha$.
- **SES**(A_{pub}, W): First compute $t = e(H_1(W), h^r) \in G_2$ for a random $r \in Z_p^*$. Output $\text{SES}(A_{pub}, W) = [g^r, H_2(t)]$.
- **Trapdoor**(A_{priv}, W): output $T_W = H_1(W)^\alpha \in G_1$.
- **Test**(A_{pub}, S, T_W): let $S = [A, B]$. Test if $H_2(e(T_W, A)) = B$. If so, output ‘yes’; if not, output ‘no’.

The proof of security relies on the difficulty of the Bilinear Diffie-Hellman problem (BDH) [Jou02].

Bilinear Diffie-Hellman Problem (BDH): Fix a generator g of G_1 . The BDH problem is as follows: given $g, g^a, g^b, g^c \in G_1$ as input, compute $e(g, g)^{abc} \in G_2$. We say that BDH is intractable if all polynomial time algorithms have a negligible advantage in solving BDH.

Theorem 3.3 [BDCOP04] The searchable encryption scheme above is semantically secure against a chosen keyword attack in the random oracle model assuming BDH is intractable.

Efficiency. The SES of [BDCOP04] is quite efficient as **KeyGen** and **Trapdoor** only need one multiplication in G_1 ; **SES** requires one multiplication in G_1 , one pairing computation, one exponentiation in G_2 and 2 accesses to the random oracle; and **Test** needs one pairing computation and one access to the random oracle.

3.1.3 Open problems

The main open problem in this area is to construct a searchable encryption in the plain model (that is without using the random oracle).

Furthermore, searchable encryption in the broader sense of searching encrypted plaintext would be very interesting to have. It would be ideal if tokens could be issued to search encrypted text on a huge server for key words and even more if one could make use of operators in this search as proposed in [Des93]. However, we think that this is a rather hard problem. So far even in the area of symmetric key cryptography no fully satisfying solution was found.

Acknowledgments

Contributors from ECRYPT: Tanja Lange, Giuseppe Persiano, Ivan Visconti.
Thanks to Yvo Desmedt.

References

- [BDCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In C. Cachin and J. Camenish, editors, *Advances in Cryptology – Eurocrypt 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer Verlag, 2004.
- [Des93] Yvo Desmedt. Computer security by redefining what a computer is. In J. B. Michael, V. Ashby, and C. Meadows, editors, *Proceedings New Security Paradigms II Workshop*, pages 160–166. ACM-SIGSAC, IEEE Computer Society Press, 1992–1993.

- [Jou02] A. Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. In *Proc. Fifth Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science. Springer Verlag, 2002.
- [MRS⁺99] P. Maniatis, M. Roussopoulos, E. Swierk, K. Lai, G. Appenzeller, X. Zhao, and M. Baker. The Mobile People Architecture. *ACM Mobile Computing and Communications Review*, (3), July 1999.

3.2 Plaintext aware encryption

3.2.1 Description and Motivations

Intuitively, an encryption scheme is *plaintext aware* (PA encryption scheme) if the only way that an adversary can produce a valid ciphertext is to apply the encryption algorithm to the public key and a message. Plaintext awareness is the strongest known form of encryption. In particular, it immediately implies security against adaptive chosen-ciphertext attack and appears to be strictly stronger.

The notion of PA encryption was first suggested by Bellare and Rogaway [BR94] with the motivation that a scheme that was PA and secure against CCA1 attacks would also be secure against CCA2 attacks. Bellare and Rogaway [BR94] provided an implementation in the Random Oracle model.

Formalization in the random oracle model Let us review the notion of an encryption scheme in the random oracle model. An encryption scheme in the random oracle model is a triplet of probabilistic polynomial time $\beta = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. A pair of public-private keys (pk, sk) is generated by running the key generator algorithm on input the security parameter 1^k . The encryption (and the decryption) algorithm has access to a random oracle H and computes the ciphertext c (the cleartext m) on input a cleartext m (a ciphertext c) and the public key pk (and the secret key sk).

Here we report the original definition of Bellare et al. [BDPR98].

Definition 3.4 [Plaintext Awareness] [BDPR98] Let $\beta = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme in the random oracle model, let B be any polynomial-time adversary and K a polynomial-time “extractor” algorithm. For any integer k , define

$$\text{SUCC}_{\beta, B, K}(k) \stackrel{\text{def}}{=} \Pr[H \leftarrow \text{HASH}; (pk, sk) \leftarrow \mathcal{K}(1^k); (hH, C, y) \leftarrow \text{View} B^{H, \mathcal{E}^H(pk)}(pk) : K(hH, C, y, pk) = D^H(sk, y) \wedge y \notin C].$$

where

1. H is the random oracle;
2. C is the list of answers received by the adversary B from the encryption oracle $\mathcal{E}^H(pk)$;
3. hH is the history of oracle queries made by B along with the answers received;

4. $\text{View}_{B^{H, \mathcal{E}^H}}(pk)$ is the view of the adversary B on input public key pk and when given access to random oracle H and encryption oracle $\mathcal{E}^H(pk)$.

We say that β is *secure in the plaintext aware sense* (PA-secure in short) if for any adversary B there exists an extractor algorithm K such that

$$\text{SUCC}_{\beta, B, K}(k) \geq 1 - \nu(k)$$

for some negligible function $\nu(\cdot)$.

Roughly speaking, the definition of PA-secure encryption scheme requires that for every adversary that manages to produce a valid ciphertext (the string y in the definition) there exists an efficient extractor that on input the public key pk , and the list of queries made by B to the oracles and without knowledge of the secret key sk outputs the corresponding cleartext. This means that any adversary B that manages to produce a valid ciphertext has knowledge of the corresponding cleartext.

Formalization without random oracle We stress that in the definition of PA presented in Section 3.2.1 the random oracle plays a crucial role. Indeed if the encryption algorithm does not access the random oracle then any decrypting adversary could extract the plaintext from a ciphertext and thus the encryption scheme would not be secure. Instead, the extractor has access to the list of queries made during the encryption and this information (which is not available to an adversary) make the extractor capable of obtaining the cleartext. Thus it seems that the random oracle is necessary for obtaining PA. We next review two recent attempts to formalize and construct PA encryption schemes without resorting to the random oracle model.

PA via key registration. In [HLM03], Herzog, Liskov and Micali presented the following simple model for PA: encryption is available only between users that have properly registered their public keys with a registration authority and the encryption is guaranteed to be PA-secure if the authority is honest.

They also propose the following elegant implementation. During the registration a sender simply gives a zero-knowledge proof of knowledge of his secret key. Since the proof system is zero-knowledge, the registration authority (be it trusted or malicious) gains no information. The encryption algorithm uses the key of both the sender and the verifier: a message is encrypted twice, once with the key of the sender and once with the key of the verifier; then a non-interactive zero-knowledge proof of correctness is appended to the ciphertext. The extractor works in the following way. During the registration phase, the extractor obtains the secret key of the sender using the extractor of the interactive zero-knowledge proof of knowledge and uses the key to obtain the cleartext.

PA in the plain model. A more recent attempt at defining the notion of a PA-secure encryption scheme has been made by Bellare and Palacio [BP04]. Here the problem outlined above that makes PA-secure encryption impossible without a random oracle is avoided by having the extractor read the coin tosses used by the ciphertext generator during the encryption procedure.

3.2.2 State of the art

In this section we review the main results concerning plaintext awareness.

Relations to other notions of security in the random oracle model In the random oracle the notion of encryption scheme secure in plaintext-aware sense is the strongest notion of security known for public-key cryptosystems.

Intuitively, security against CCA2 attack is achieved using a semantically secure encryption scheme [GM84] whose encryption algorithm also produces a non-interactive zero-knowledge proof of knowledge of the cleartext. This makes the system secure as the decryption oracle can only be queried on ciphertexts for which the adversary has proved knowledge of the corresponding cleartexts thus making the decryption oracle useless for the adversary. Then, if one has an encryption scheme secure in the plaintext-aware sense then there is no need to produce a non-interactive zero-knowledge proof as, knowledge of the cleartext, is guaranteed by the plaintext-aware security. This intuition is formalized by the following result.

Theorem 3.5 [BDPR98] Let $\beta = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be any encryption scheme secure in the plaintext aware sense in the random oracle model. Then β is CCA2-secure in the random oracle model.

It is also easy to see that not all CCA2-secure encryption schemes are secure in the plaintext aware sense by the following argument. Indeed take any CCA2-secure encryption scheme and modify it so that the key generation algorithm outputs the encryption of a randomly generated cleartext. Clearly the modified scheme is still CCA2-secure but is not secure in the plaintext aware sense.

Theorem 3.6 [BDPR98] If there exists a CCA2-secure encryption scheme then there exists a CCA2-secure encryption scheme that is not secure in the plaintext aware sense.

Constructions A natural approach to designing PA-secure encryption schemes is based on the use of non-interactive zero-knowledge proofs of knowledge [DSP92] (NIZK-PoK, in short). One possible construction is the following. Let $\beta = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a semantically secure encryption scheme and consider an augmented encryption scheme in which a public key has the form (pk, R) where pk is the public key of β and R is a random string. The encryption algorithm simply consists of running algorithm \mathcal{E} of the “base” encryption scheme and then computing an NIZK-PoK of the cleartext using R as reference string. The intuition is to use the extractor of the NIZK PoK to extract the cleartext from the proof of knowledge found in the ciphertext. The problem with this approach is that the PA extractor must work with a given public key and thus cannot choose the random reference string R . Unfortunately, the ability to choose the random reference string is crucial for the extractor of the PoK.

Currently, as proved by [BP04], the most efficient construction of a PA-secure encryption scheme is a scheme due to D amgaard denoted DEG [Dam91]. The security of the DEG encryption scheme is based on the so-called Diffie-Hellman Knowledge assumption which is discussed below.

The DHK assumption. Let G be a prime-order group of order $p = 2q + 1$, where q is a prime, and let g be a generator for G . We say that the triple (A, B, W) is a *DH-triple* if there exist $a, b \in \mathbb{Z}_q$ such that $A = g^a$, $B = g^b$ and $W = g^{ab}$ (all operations are in G). Moreover we say that (B, W) is a *DH-pair relative to A* if (A, B, W) is a DH-triple. One way for an adversary to construct a DH-pair (B, W) relative to a given A is to pick some $b \in \mathbb{Z}_q$, set $B = g^b$ and $W = A^b$, and output (B, W) . The DHK assumption informally says that this is the only way that a polynomial-time adversary can output a DH-pair relative to a given A . This is formalized by assuming that for any adversary that manages to compute a DH-pair (B, W) relative to A , there exists an extractor that outputs the discrete log to the base g of B modulo p .

3.2.3 Open problems

The current state of knowledge presents two implementations (besides the ones in the random oracle model) of PA-secure encryption schemes. Both suffer of some limitations that we would like to see overcome.

The construction of [HLM03] departs from the widely-accepted public-key model since it requires both the sender and receiver to have registered their keys. Moreover, it is assumed that there exists a trusted registration authority that acts as a verifier of knowledge each time a new key is registered. On the positive side, the construction of [HLM03] is based on very general assumptions.

The cryptosystem DEG, due to Damgård [Dam91], and proved PA-secure in [BP04] assumes the standard model for public-key cryptography. On the other hand it is based on a very strong and quite non-standard assumption.

The following is an interesting open problem.

Open Problem 1 *Give a construction in the standard public-key model of a PA-secure encryption scheme. Base the construction on general or widely used cryptographic assumption like, for example, existence of a semantically secure encryption scheme, security of El Gamal encryption scheme, quadratic residuosity assumption.*

Acknowledgments

Contributors from ECRYPT: Giuseppe Persiano, Ivan Visconti.

References

- [BDPR98] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology – Crypto ’98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–46. Springer Verlag, 1998.

- [BP04] M. Bellare and A. Palacio. Towards plaintext-aware public-key encryption without random oracles. In P. J. Lee, editor, *Advances in Cryptology – ASIACRYPT ’04*, volume 3329 of *Lecture Notes in Computer Science*, page ?? Springer Verlag, 2004.
- [BR94] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. De Santis, editor, *Advances in Cryptology – EuroCrypt ’94*, volume 950 of *Lecture Notes in Computer Science*. Springer Verlag, 1994.
- [Dam91] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum, editor, *Advances in Cryptology – Crypto ’91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer Verlag, 1991.
- [DSP92] A. De Santis and G. Persiano. Zero-knowledge proofs of knowledge without interaction. In *Proceedings of the 33rd Symposium on Foundations of Computer Science 1992, (FOCS ’92)*, pages 427–437, 1992.
- [GM84] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
- [HLM03] J. Herzog, M. Liskov, and S. Micali. Plaintext awareness via key registration. In D. Boneh, editor, *Advances in Cryptology – Crypto ’03*, volume 2729 of *Lecture Notes in Computer Science*, pages 548–564. Springer Verlag, 2003.

3.3 Verifiable encryption

3.3.1 Introduction

A *verifiable encryption scheme* is in its basic form a two-party protocol between a prover P and a verifier V . Their common inputs are a public encryption key E , a public value x , and a binary relation \mathcal{R} . As a result of the protocol, V either rejects or obtains the encryption of some value w under E such that $(x, w) \in \mathcal{R}$ holds. For instance, \mathcal{R} could be defined such that $(x, w) \in \mathcal{R}$ if and only if w is a signature on message x w.r.t. to some fixed public key. In other words, P claims to have given V the encryption of a valid signature on x .

The protocol should ensure that V accepts an encryption of an invalid w with only negligible probability. Moreover, V should learn nothing except the encryption of w and the fact that w is valid w.r.t. x . In particular, if the encryption scheme is semantically secure, the protocol should be zero-knowledge.

The encryption key E can belong to P , but typically belongs to a third party in which case the third party should not need to take part in the protocol, in other words, P does not need to know the secret key corresponding to E .

Verifiable encryption schemes are employed in many cryptographic protocols (although the term “verifiable encryption” is not always used). Examples are digital payment systems with revocable anonymity (e.g., [CMS96, FTY96]), verifiable signature sharing (e.g., [FR95]), (publicly) verifiable secret sharing (e.g., [Sta96]), escrow schemes [PS00, YY98], or fair exchange of signatures [Ate99, ASW98, Bao00]. We discuss some of these applications below.

However, only the schemes presented in [ASW98, KP98, SPC95] do not apply ad-hoc constructions using a specific encryption scheme that suits the particular application. Also, most of the ad-hoc constructions neglect security against chosen ciphertext attacks, which in most application is a crucial requirement (cf. our discussion of the applications below).

Micali [Mic] also proposes the use of provable encryption of data for third parties to solve several variants of the fair exchange problem.

3.3.2 Applications

In this section, we outline some of the numerous applications of verifiable encryption [CS02].

Key Escrow. Party A may encrypt its own secret key for an asymmetric cryptographic primitive under the public key of a trusted third party T , and present to a second party B the ciphertext ψ and a proof that ψ is indeed an encryption of its secret key. This problem area has attracted a good deal of attention, with specific schemes being proposed in [Sta96, BG96, YY98, ASW00, PS00].

In this application, user A would attach a label to ψ that indicates the conditions under which ψ should be decrypted, e.g., A 's identity and perhaps an expiration date. The definition of chosen ciphertext security ensures that decrypting a ciphertext under any label different from the label used to create the ciphertext reveals no information about the original encrypted message.

Optimistic Fair Exchange Two parties A and B want to exchange some valuable digital data (e.g., signatures on a contract, e-cash), but in a fair way: either each party obtains the other's data, or neither party does. One way to do this is by employing a trusted third party T , but, for the sake of efficiency, with T only involved in crisis situations. One approach to this problem is to have both parties verifiably encrypt to each other their data under T 's public key, and only then to reveal their data to each other — if one party backs out unexpectedly, the other can go to T to obtain the required data. The general problem of optimistic fair exchange has been extensively studied, c.f., [ASW97, BDM98, BP90, Mic, ASW00], while the solution using verifiable encryption was studied in detail in [ASW00].

As in the escrow application, the label mechanism plays a crucial role here, helping to enforce the logic of the exchange protocol, and a verifiable decryption protocol may be used to hold T 's feet to the fire.

Publicly Verifiable Secret Sharing and Signature Sharing. Stadler [Sta96] introduced the notion of *publicly verifiable secret sharing*. Here, one party, the dealer, shares a secret with several proxies P_1, \dots, P_n , in such a way that a third party (other than the dealer and the proxies) can verify that the sharing was done correctly. This can be done quite simply by sharing the secret using Shamir's secret sharing scheme: the dealer encrypts P_i 's share under P_i 's public key, and gives to the third party commitments to these shares, along with commitments to the coefficients of the blinding polynomial, and all of the ciphertexts, and proves to the third party that the ciphertexts encrypt openings of the commitments to the

shares. As the openings to the commitments are just discrete logarithms, verifiable encryption of discrete logarithms is just the right tool.

Universally Composable Commitments. The notion of *universally composable (UC) commitments*, introduced by Canetti and Fischlin [CF01], is a very strong notion of security for a commitment scheme. It basically says that commitments in the real world act like commitments in an ideal world in which, when a party A commits to a value x to a party B , A presents x to an idealized trusted party T (that does not exist in the real world), and when A opens the commitment, T gives x to B . In the ideal world, no information about x is revealed to B prior to opening, and A is forced to fix the value committed to when the commitment protocol runs.

This notion of security is so strong, in fact, that it can only be realized in the *common reference string (CRS)* model, where all parties have access to a string that was generated by a trusted party according to some prescribed distribution. In the CRS model, the simulator S in the ideal world is given the privilege of generating the common reference string, and so S may know some “side information” related to the common reference string that is not available to anyone in the real world.

Verifiable encryption of a representation may be used to implement UC commitments in the CRS model, as follows. The CRS consists of a public key for the encryption scheme, along with bases γ_1 and γ_2 for some suitable group. When A commits a value x to B , he creates a Pedersen commitment $C = \gamma_1^x \gamma_2^r$, and an encryption ψ of the representation (x, r) of C with respect to (γ_1, γ_2) . A then gives (C, ψ) to B , and proves to B that ψ indeed decrypts to a representation of C . In order to satisfy the definition of security for UC commitments, and in particular, to prevent “man in the middle attacks,” a label containing A ’s identity should be attached to ψ .

The reason this is secure is that the simulator S in the CRS model knows the secret key to the encryption scheme, which allows him to “extract” values committed by corrupted parties, and S knows the discrete logarithm of γ_2 with respect to γ_1 , which allows him to “equivocate” values committed by honest parties. The proof that ψ is an encryption of a representation C ensures that the value extracted by the simulator at commitment time agrees with the value revealed at opening time.

Confirmer Signatures In a confirmer signature scheme, a notion introduced in [Cha94], a party A creates an “opaque signature” ψ on a message m , which can not be verified by any other party except a designated trusted third party T , who may either confirm or deny the validity of the signature to another party B . Under appropriate circumstances, T may also *convert* ψ into an ordinary signature, which may then be verified by anybody. Additionally, the party A may prove the validity of an opaque signature ψ to a party B , at the time that A creates and gives ψ to B . As described in [CM00], one may implement confirmer signatures as follows: A creates an ordinary signature σ on m , and encrypts σ under T ’s public key. Using verifiable encryption, A may prove to B that the resulting ciphertext ψ indeed encrypts a valid signature on m , and using verifiable decryption, T may confirm or deny the validity of ψ , or alternatively, just decrypt ψ , thus converting it to the ordinary signature σ .

Group Signatures and Anonymous Credentials In a group signature scheme (see [ACJT00, KP98, CD00]), when a user joined a group (whose membership is controlled by a special party, called the *group manager*), the user may sign messages on behalf of the group, without revealing his individual identity; however, under appropriate circumstances, the identity of the individual who actually signed a particular message may be revealed (using a special party, called the *anonymity revocation manager*, which may be distinct from the group manager).

Without going into too many details, verifiable encryption may be used in the following way as a component in such a system. When a group member signs a message, he encrypts enough information under the public key of the anonymity revocation manager, so that later, if the identity of the signer needs to be revealed, this information can be decrypted. To prove that this information correctly identifies the signer, he makes a Pedersen commitment to this information, proves that the committed value identifies the user, encrypts the opening of the commitment, and proves that the ciphertext decrypts to an opening of the commitment. To turn this into a signature scheme, one must use the Fiat-Shamir heuristic [FS87] to make it non-interactive (the interactive version is called an *identity escrow* scheme [KP98]).

Although one can implement group signatures without it, by using verifiable encryption, one can build a more modular system, in which the group manager and anonymity manager are separate entities with independently generated public keys (this is the separability issue). Verifiable decryption can be used both to ensure the correct behavior of the anonymity revocation manager (preventing it from “framing” innocent users), and to allow even more fine-grained control of anonymity revocation: instead of simply revealing the identity of a particular signer, the anonymity revocation manager can state (and prove) whether or not a particular signature was generated by a particular user.

Credential systems [Cha85, CL01] are a generalization of group signatures that allow users to show credentials to various organizations, and obtain new credentials, without revealing their identity, except through the use of an anonymity revocation manager. Verifiable encryption can be used as a component in such systems in a manner similar to that described above for group signatures.

3.3.3 Security Definition

This section recalls the definition of verifiable encryption given by Camenisch and Shoup [CS03].

Before stating the formal definition of verifiable encryption, we begin with a high level discussion of what we are after, along with some auxiliary definitions.

Let $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a public key encryption scheme, and suppose we have generated a key pair (PK, SK) .

A verifiable encryption scheme proves that a ciphertext encrypts a plaintext satisfying a certain relation \mathcal{R} . The relation \mathcal{R} is defined by a *generator* algorithm \mathcal{G}' which on input 1^λ outputs a *description* $\Psi = \Psi[\mathcal{R}, W, \Delta]$ of a binary relation \mathcal{R} on $W \times \Delta$. We require that the sets \mathcal{R} , W , and Δ are easy to recognize (given Ψ). For $\delta \in \Delta$, an element $w \in W$ such that $(w, \delta) \in \mathcal{R}$ is called a *witness* for δ . The idea is that the encryptor will be given a value δ , a

witness w for δ , and a label L , and then encrypts w under L , yielding a ciphertext ψ . After this, the encryptor may prove to another party that ψ decrypts under L to a witness for δ . In carrying out the proof, the encryptor will of course need to make use of the random coins that were used by the encryption algorithm: we denote by $\mathcal{E}'(PK, m, L)$ the pair (ψ, coins) , where ψ is the output of $\mathcal{E}(PK, m, L)$ and coins are the random coins used by \mathcal{E} to compute ψ .

In such a proof system, the (honest) verifier will output 0 or 1, with 1 signifying “accept.” We of course shall require that the proof system is sound, in the sense that if a verifier accepts a proof, then with overwhelming probability, ψ indeed decrypts under L to a witness for δ . However, it is convenient, and adequate for many applications, to take a more relaxed approach: instead of requiring that ψ decrypts under L to a witness, we only require that a witness can be easily reconstructed from the plaintext using some efficient *reconstruction* algorithm. Such an algorithm *recon* takes as input a public key PK , a relation description $\Psi[\mathcal{R}, W, \delta]$, an element $\delta \in \Delta$, and a message $m \in M_{PK} \cup \{\text{reject}\}$, and outputs $w \in W \cup \{\text{reject}\}$.

We need to make some technical “compatibility” requirements: we say that an encryption scheme, a relation generator, and a reconstruction algorithm as above are *mutually compatible* if for all $\lambda \geq 0$, all $(PK, SK) \in \mathcal{G}(1^\lambda)$, and all $\Psi[\mathcal{R}, W, \Delta] \in \mathcal{G}'(1^\lambda)$, we have

- $W \subset M_{PK}$, and
- for all $(w, \delta) \in \mathcal{R}$, we have $\text{recon}(PK, \Psi, \delta, w) = w$.

The first requirement simply says that witness “fit” into the message space, and the second requirement simply says that the reconstruction routine does not modify valid witnesses (together with the correctness property for the encryption scheme, this ensures that an encryption of a witness decrypts and reconstructs to the same witness).

We shall also require that the proof system is special honest-verifier zero knowledge. To formulate this more precisely below, we let $\text{Trans}(PK, \Psi, \delta, \psi, L, c, w, \text{coins})$ denote the transcript seen by a verifier that uses a *fixed* challenge c .

Definition 3.7 A proof system $(\mathcal{P}, \mathcal{V})$, together with mutually compatible encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, relation generator \mathcal{G}' , and reconstruction algorithm *recon*, form a *verifiable encryption scheme*, if the following properties hold.

Correctness: for all $(PK, SK) \in \mathcal{G}(1^\lambda)$, for all $\Psi[\mathcal{R}, W, \Delta] \in \mathcal{G}'(1^\lambda)$, for all $(w, \delta) \in \mathcal{R}$, for all $L \in \{0, 1\}^*$, for all $(\psi, \text{coins}) \in \mathcal{E}'(PK, w, L)$,

$$\Pr[x \leftarrow \mathcal{V}(PK, \Psi, \delta, \psi, L)_{\mathcal{P}(PK, \Psi, \delta, \psi, L, w, \text{coins})} : x = 1] = 1 - \text{neg}(\lambda).$$

Soundness: for all adversaries $(\mathcal{A}^*, \mathcal{P}^*)$,

$$\Pr[\begin{array}{l} (PK, SK) \leftarrow \mathcal{G}(1^\lambda); \Psi[\mathcal{R}, W, \Delta] \leftarrow \mathcal{G}'(1^\lambda); \\ (\delta, \psi, L, \text{aux}) \leftarrow \mathcal{A}^*(PK, SK, \Psi); \\ x \leftarrow \mathcal{V}(PK, \Psi, \delta, \psi, L)_{\mathcal{P}^*(\text{aux})}; \\ m \leftarrow \mathcal{D}(SK, \psi, L); \\ w \leftarrow \text{recon}(PK, \Psi, \delta, m); \\ x = 1 \wedge (w, \delta) \notin \mathcal{R} \end{array}] = \text{neg}(\lambda).$$

Special honest-verifier zero knowledge: There exists a simulator Sim such that for all adversaries $(\mathcal{A}^*, \mathcal{B}^*, \mathcal{C}^*)$, we have

$$\begin{aligned}
 \Pr[& (PK, SK) \leftarrow \mathcal{G}(1^\lambda); \Psi[\mathcal{R}, W, \Delta] \leftarrow \mathcal{G}'(1^\lambda); \\
 & (w, \delta, L, aux) \leftarrow \mathcal{A}^*(PK, SK, \Psi), \text{ where } (w, \delta) \in \mathcal{R}; \\
 & (\psi, coins) \leftarrow \mathcal{E}'(PK, w, L); \\
 & c \leftarrow \mathcal{B}^*(aux, \psi); \\
 & b \leftarrow \{0, 1\}; \\
 & \text{if } b = 0 \\
 & \quad \text{then } \alpha \leftarrow Trans(PK, \Psi, \delta, \psi, L, c, w, coins) \\
 & \quad \text{else } \alpha \leftarrow Sim(PK, \Psi, \delta, \psi, L, c); \\
 & \hat{b} \leftarrow \mathcal{C}^*(aux, \psi, \alpha) : \\
 & b = \hat{b} \qquad \qquad \qquad] = 1/2 + \text{neg}(\lambda).
 \end{aligned}$$

The above definitions are fairly traditional. Our formulations of soundness and special honest-verifier zero knowledge are basically of the “computational” variety, but where we have taken the notion of “computational” one step further: instead of universally quantifying over the inputs to the verifier (respectively, simulator), we quantify “computationally.” This is technically convenient, and is adequate for most applications.

Also, the above definitions assume that the key for the encryption scheme are generated by a trusted party. While it is possible to define verifiable encryption in a setting where the keys are not generated by a trusted party, the definitions in this case are a bit more complicated and subtle, and we do not present them here. Nevertheless, our protocols would require only slight modification to remain secure in this setting.

3.3.4 Proposed Schemes

The concept of verifiable encryption was introduced in [Sta96] in the context of publicly verifiable secret sharing schemes, and in a more general form in [ASW98], for the purpose of fair exchange of signatures.

Stadler [Sta96] proposed a scheme verifiably encrypting discrete logarithms for the ElGamal encryption scheme using on so-called double discrete logarithms. Stadler’s protocol can be applied to the Cramer-Shoup encryption scheme [CS98] to obtain chosen ciphertext security. Asokan et al. [ASW98] proposed a scheme that works any encryption scheme but only for relations \mathcal{R} containing pairs of form $(x, f^{-1}(x))$, where f is a one-way group homomorphism and the verifier is required store k encryptions of the underlying encryption scheme. Camenisch and Damgård generalized the protocols employed in [ASW98, KP98, SPC95] and present a verifiable encryption scheme that works for any encryption scheme and for relations \mathcal{R} that possess a three-move zero-knowledge proof of knowledge which is an Arthur-Merlin game, i.e., as the second message, the verifier sends a random challenge.

All these scheme mentioned above employ three moves proofs of knowledge with binary-challenges and therefore are usually not very efficient. Camenisch and Shoup [CS02, CS03] provide a variant of the Paillier encryption scheme [Pai99] together with efficient zero-knowledge protocols to prove that a given ciphertext is the encryption of a discrete logarithm.

3.3.5 Open Problems

The schemes that are constructed along the lines of [CD00, ASW98, KP98, SPC95] work for any encryption scheme but can only use binary challenges for the three-move protocols for the relation \mathcal{R} , although the protocols themselves would allow for larger challenges. It is an open problem to find a method to obtain a verifiable encryption schemes that can use larger challenges which would make the resulting scheme more efficient.

Another open question is to find an alternative encryption scheme (i.e., one based on an other computational assumption) to the one by Camenisch and Shoup that allows one to encrypt discrete logarithms and then subsequently to prove that one encrypted indeed the discrete logarithm of some given group element.

Finally, it would be nice to find an encryption scheme that allows one to encrypt witnesses of relations other than discrete logarithm based one and then to efficiently prove that one indeed encrypted such a witness.

Acknowledgments

Contributors from ECRYPT: Jan Camenisch.

References

- [ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Mihir Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer Verlag, 2000.
- [ASW97] N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for fair exchange. In *Proc. 4th ACM Conference on Computer and Communications Security*, pages 6–17, 1997.
- [ASW98] Nadarajah Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. In Kaisa Nyberg, editor, *Advances in Cryptology — EURO-CRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606. Springer Verlag, 1998.
- [ASW00] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):591–610, April 2000.
- [Ate99] Giuseppe Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *Proc. 6th ACM Conference on Computer and Communications Security*, pages 138–146. ACM press, November 1999.
- [Bao00] Feng Bao. An efficient verifiable encryption scheme for the encryption of discrete logarithms. In Jean-Jaques Quisquater and Bruce Schneier, editors, *Smart Card*

- Research and Applications (CARDIS '98)*, volume 1820 of *Lecture Notes in Computer Science*. Springer Verlag, 2000.
- [BDM98] Feng Bao, Robert Deng, and Wenbo Mao. Efficient and practical fair exchange protocols with off-line TTP. In *IEEE Symposium on Security and Privacy*, pages 77–85. IEEE Computer Society Press, 1998.
- [BG96] Mihir Bellare and Shafi Goldwasser. Encapsulated key escrow. Technical Report TR 688, MIT Laboratory for Computer Science, April 1996.
- [BP90] Holger Bürk and Andreas Pfitzmann. Digital payment systems enabling security and unobservability. *Computer & Security*, 9(8):715–721, 1990.
- [CD00] Jan Camenisch and Ivan Damgård. Verifiable encryption, group encryption, and their applications to group signatures and signature sharing schemes. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer Verlag, 2000.
- [CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 19–40. Springer Verlag, 2001.
- [Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.
- [Cha94] David Chaum. Designated confirmer signatures. In Alfredo De Santis, editor, *Advances in Cryptology — EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 86–91. Springer Verlag Berlin, 1994.
- [CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer Verlag, 2001.
- [CM00] Jan Camenisch and Markus Michels. Confirmer signature schemes secure against adaptive adversaries. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 243–258. Springer Verlag, 2000.
- [CMS96] Jan Camenisch, Ueli Maurer, and Markus Stadler. Digital payment systems with passive anonymity-revoking trustees. In Elisa Bertino, Helmut Kurth, Giancarlo Martella, and Emilio Montolivo, editors, *Computer Security — ESORICS 96*, volume 1146 of *Lecture Notes in Computer Science*, pages 33–43. Springer Verlag, 1996.
- [CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, volume 1642 of *Lecture Notes in Computer Science*, pages 13–25, Berlin, 1998. Springer Verlag.
- [CS02] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. <http://eprint.iacr.org/2002/161>, 2002.

- [CS03] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, Lecture Notes in Computer Science, 2003.
- [FR95] Matthew Franklin and Michael Reiter. Verifiable signature sharing. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology — EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 50–63. Springer Verlag, 1995.
- [FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Verlag, 1987.
- [FTY96] Yair Frankel, Yiannis Tsiounis, and Moti Yung. “Indirect discourse proofs:” Achieving efficient fair off-line E-cash. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology — ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 286–300. Springer Verlag, 1996.
- [KP98] Joe Kilian and Erez Petrank. Identity escrow. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, volume 1642 of *Lecture Notes in Computer Science*, pages 169–185, Berlin, 1998. Springer Verlag.
- [Mic] Silvio Micali. Efficient certificate revocation and certified E-mail with transparent post offices. Presentation at the 1997 RSA Security Conference.
- [Pai99] Pascal Paillier. Public-key cryptosystems based on composite residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–239. Springer Verlag, 1999.
- [PS00] Guillaume Poupard and Jacques Stern. Fair encryption of RSA keys. In Bart Preneel, editor, *Advances in Cryptology: EUROCRYPT 2000*, volume 1087 of *Lecture Notes in Computer Science*, pages 173–190. Springer Verlag, 2000.
- [SPC95] Markus Stadler, Jean-Marc Piveteau, and Jan Camenisch. Fair blind signatures. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology — EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 209–219. Springer Verlag, 1995.
- [Sta96] Markus Stadler. Publicly verifiable secret sharing. In Ueli Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 191–199. Springer Verlag, 1996.
- [YY98] Adam Young and Moti Young. Auto-recoverable auto-certifiable cryptosystems. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 17–31. Springer Verlag, 1998.

4 Signcryption

4.1 Description and motivation

4.1.1 Introduction

Encryption and signature schemes are the basic tools offered by public key cryptography for providing privacy and authenticity respectively. For a long time they were viewed as important, but distinct, building blocks for higher level protocols; however, there are many settings where both are needed, perhaps the most obvious being secure e-mailing. In this scenario messages should be encrypted to ensure confidentiality and signed to provide authentication. In this case it is of course possible to use an encryption scheme combined with a digital signature scheme. However, as has been observed by An, Dodis and Rabin [ADR02], there are often subtleties when doing this. Moreover, it may be possible to use features particular to the case where one wants both authentication and encryption to gain in efficiency and functionality. Motivated by such considerations there has been much recent research into schemes and methods for simultaneous signing and encrypting.

The first proposed construction combining the functionality of an encryption scheme with that of a signature scheme appeared in a paper by Zheng [Zhe97]. The motivation behind this work was to obtain some efficiency benefit when compared to encrypting and signing separately. Zheng called schemes designed to achieve this goal *signcryption* schemes. Many schemes have subsequently been designed with Zheng's original motivation in mind [BSZ02, BD98, LQ, ML04, MLM03, PM98]. Some of these have been formally analysed using complexity-theoretic reductions [BSZ02, LQ, ML04, MLM03]; however, all this analysis relies on the random oracle model [BR93].

The first time that a formal security treatment was applied to signcryption schemes was the work of An et al. [ADR02]. Unlike the work of Zheng, the purpose here was not simply to achieve efficiency: the goal was to provide a rigorous framework to analyse any scheme or composition paradigm used to achieve the combined functionality of encryption and signature. Several security notions were introduced: *insider* and *outsider* security; and *two-user* and *multi-user* security. We will discuss these notions further in the next section. In addition to providing a security framework, several composition paradigms are also proposed and analysed in [ADR02].

4.1.2 Signcryption schemes

A signcryption scheme S consists of the following algorithm.

- The sender key generation algorithm \mathcal{K}_s is randomised. It takes as input a security parameter 1^k and returns a matching public/secret key pair (pk_s, sk_s) for the sender.
- The receiver key generation algorithm \mathcal{K}_r is randomised. It takes as input a security parameter 1^k and returns a matching public/secret key pair (pk_r, sk_r) for the receiver.

- The signcryption algorithm \mathcal{S} is randomised. It takes as input a sender's keys (pk_s, sk_s) , a receiver's public key pk_r and a plaintext m . It returns a ciphertext σ .
- The unsigncryption algorithm \mathcal{U} is deterministic. It takes as input a sender's public key pk_s , a receiver's keys (pk_r, sk_r) , and a string σ . It returns either a message m or the symbol \perp . The symbol \perp indicates that the signcryption was invalid.

Note that, unlike a digital signature scheme, a signcryption does not support non-repudiation of messages by default. The reason for this is that we are dealing with encrypted data and so only the intended receiver of a signcryption can perform the unsigncryption operation.

4.2 State of the art

4.2.1 Security models

The accepted definitions of security for signcryption schemes come from [ADR02]. Let us first consider the basic definition pertaining to confidentiality. This says that, given keys pk_s and pk_r , an adversary should not be able to distinguish between signcryptions of two chosen messages. In its attempt to distinguish an adversary is given oracle access to \mathcal{S} that will produce signcryptions using (pk_s, sk_s) and pk_r . It is also given access oracle access to \mathcal{D} that will perform unsigncryption using pk_s and (pk_r, sk_r) .

The basic definition of unforgeability for signcryption is similar: an adversary given pk_s and pk_r cannot produce a valid signcryption using access to the oracles described above.

There are several ways in which the definitions sketched above can be strengthened. We outline these below. Motivation for considering any of these definitions may be found in [ADR02].

outsider security The definitions that we have given so far are instances of outsider security: in the case of confidentiality, an adversary is assumed to have access to an oracle \mathcal{S} that uses (pk_s, sk_s) and pk_r ; however, it is an *outsider* of the system in that it does not know sk_s . (If it knew sk_r it could perform decryption itself and so indistinguishability of signcryptions would be impossible!)

insider security Note that if we surrender sk_s to an adversary we obtain a standard public-key encryption scheme. This is so because sk_s can be used by the adversary to produce ciphertexts. This resulting scheme is called the *induced* encryption scheme in [ADR02]. The construction is said to offer insider security – with respect to confidentiality – if the induced encryption scheme is IND-CCA2 secure [RS92]. An analogous definition for unforgeability given knowledge of the receiver's secret key is formulated in [ADR02].

two-user setting The definitions that we have discussed thus far are instances of security in a two-user setting. For example, an adversary wishes to distinguish signcryptions

created using (pk_s, sk_s) and pk_r . In its attack the adversary has a signcryption oracle that uses these keys together with an unsigncryption oracle that uses pk_s and (pk_r, sk_r) . The point to observe here is that these oracles used fixed keys.

multi-user setting Unlike in the two-user setting, in a multi-user setting the adversary is able to choose the public keys to input to its oracles. For example, it can obtain signcryptions produced using (pk_s, sk_s) and an arbitrary public key of its choosing.

4.2.2 Schemes

In this section we restrict our attention to schemes that offer proofs of security.

Generic constructions In [ADR02], several constructions are analysed. These constructions all require an encryption scheme that is IND-CCA2 secure [RS92] and a signature scheme that is existentially unforgeable under adaptive chosen message attack [GMR88]. Although these constructions offer proofs of security, they do not meet Zheng’s original goal: to obtain some efficiency benefit when compared to encrypting and signing separately.

Discrete logarithm based constructions The original construction of Zheng is very efficient: one group exponentiation for signcryption and three for unsigncryption [Zhe97]. An equally efficient variant of the scheme was subsequently proved secure by Baek et al. [BSZ02]. (The scheme does not provide insider security with respect to confidentiality.) The disadvantage with the scheme is that there is no obvious way to provide non-repudiation. The only secure method suggested requires highly non-trivial zero-knowledge proofs. No suggestion of how to implement these is given.

A solution without the non-repudiation shortcoming of Zheng’s scheme is given by Malone-Lee [ML04]. This is achieved at the cost of an extra group exponentiation for the signcryption operation. Malone-Lee also shows that, by using a gap-Diffie-Hellman group [OP01], an improved security reduction is possible. Again, this scheme does not offer insider security with respect to confidentiality.

A second scheme using gap-Diffie-Hellman groups has been proposed by Libert and Quisquater [LQ04]. This scheme offers an additional property that the identity of the sender does not need to be known a priori by the receiver: the scheme offers *key privacy*.

Factoring based constructions Seinfeld and Zheng were the first to propose a signcryption scheme based on a hard problem related to factoring [SZ00]. This scheme has two drawbacks. First of all a trusted authority is required to publish a public modulus. Secondly, a non-standard assumption is necessary for the proofs of security.

A signcryption scheme with proofs of security under the RSA assumption was proposed by Malone-Lee and Mao [MLM03]. This scheme offers a bandwidth advantage over signing and encrypting using RSA; however, it does not offer any computational advantage.

4.3 Open problems

The only results for signcryption that do not rely on the random oracle model [BR93] are those of An et al. for the generic constructions [ADR02]. This means that one has to use an encryption scheme combined with a signature scheme, both provably secure on their own in the standard model. This is the benchmark for any future results for signcryption outside the random oracle model.

The vast majority of the known constructions today use the discrete logarithm problem for their security. Any new schemes based on RSA would be interesting.

Finally, an elaborate security model for *identity-based signcryption* has been proposed by Boyen [Boy03]. It may be worthwhile investigating adapting some of Boyen's definitions to the standard public key setting for signcryption.

Acknowledgments

Contributors from ECRYPT: John Malone-Lee.

References

- [ADR02] J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer-Verlag, 2002.
- [BD98] F. Bao and R. H. Deng. A signcryption scheme with signature directly verifiable by public key. In *Public Key Cryptography - PKC '98*, volume 1431 of *Lecture Notes in Computer Science*, pages 55–59. Springer-Verlag, 1998.
- [Boy03] X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 382–398. Springer-Verlag, 2003. Full version available at <http://eprint.iacr.org/2003/163/>.
- [BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
- [BSZ02] J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. In *Public Key Cryptography - PKC 2002*, volume 2274 of *Lecture Notes in Computer Science*, pages 80–98. Springer-Verlag, 2002.
- [GMR88] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.
- [LQ] B. Libert and J. J. Quisquater. New identity-based signcryption schemes from pairings. In *IEEE Information Theory Workshop 2003*. Full version available at <http://eprint.iacr.org/2003/023/>.

- [LQ04] B. Libert and J. J. Quisquater. Efficient signcryption with key privacy from gap Diffie-Hellman groups. In *Public Key Cryptography - PKC 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 187–200. Springer-Verlag, 2004.
- [ML04] J. Malone-Lee. Signcryption with non-interactive non-repudiation. Technical Report CSTR-02-004, Department of Computer Science, University of Bristol, 2004. Available at <http://www.cs.bris.ac.uk/Publications/index.jsp>.
- [MLM03] J. Malone-Lee and W. Mao. Two birds one stone: Signcryption using RSA. In *Topics in Cryptology - CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 211–226. Springer-Verlag, 2003.
- [OP01] T. Okamoto and D. Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In *Public Key Cryptography - PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 104–118. Springer-Verlag, 2001.
- [PM98] H. Petersen and M. Michels. Cryptanalysis and improvement of signcryption schemes. *IEE Proceedings - Computers and Digital Techniques*, 145(2):149–151, 1998.
- [RS92] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer-Verlag, 1992.
- [SZ00] R. Steinfeld and Y. Zheng. A signcryption scheme based on integer factorization. In *ISW 2000*, volume 1975 of *Lecture Notes in Computer Science*, pages 309–322. Springer-Verlag, 2000.
- [Zhe97] Y. Zheng. Digital signcryption or how to achieve $\text{cost}(\text{signature} \ \& \ \text{encryption}) \ll \text{cost}(\text{signature}) + \text{cost}(\text{encryption})$. In *Advances in Cryptology - CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.

5 Homomorphic schemes

5.1 Introduction

Despite the huge amount of research that has been carried so far in cryptography, very few convincing trapdoor mechanisms have been proposed. Many encryption schemes have been presented, but they are all, more or less, variants of the same few basic ideas. In principle it is possible to distinguish between two main types of cryptosystems: the ones based on RSA (and related assumptions) and those based on the discrete log (and related assumptions).

Of course several other constructs of different nature have been proposed, but they suffer from either inefficiency or security flaws. To this “class”, belong almost all lattice-based cryptosystems (for example [2], subsequently broken by [19]) and knapsack-type schemes (like [9] later broken by [24]). In the very end, it appears that basically all “trusted” cryptosystems are schemes either based on the RSA or the discrete log mechanisms.

In a nutshell the main difference between the two approaches can be described as follows. The schemes based on RSA and related assumptions take advantage of the fact that RSA is a (conjectured) trapdoor function. The underlying idea is then to conjugate the feasibility of extracting roots of polynomials over finite fields when the trapdoor information is available with the intractability of the same problem when such an information is not available. The discrete log, on the other hand, is conjectured to be just a one-way function, and the underlying idea of the schemes that rely on this assumption is, in general, to use Diffie-Hellman variants to securely encrypt and decrypt. A positive side of discrete-log related schemes is that they can, in general, take advantage of the *homomorphic* property of the exponentiation function. Very informally, this property, guarantees that given $f(a) = g^a$ and $f(b) = g^b$ for some public basis g , one can easily compute $f(a + b)$ without needing to know a and b explicitly. This is a very useful property especially in contexts like electronic voting, e-commerce and distributed cryptography in general.

In this sense a natural question is therefore the following: is it possible to somehow conjugate the two approaches in order to get the positive aspects of both? In other words, could one devise a trapdoor mechanism with the property of being a homomorphic function as well?

5.2 An Overview of Known Constructions

In recent years a new direction of research started to give a positive answer to the above question. The developed methods, known as *trapdoors in the discrete log*, arise from the algebraic setting of high degree residuosity classes. In such schemes, the message space is a ring \mathcal{M} of modular residues and ciphertexts are in the group \mathcal{G} (denoted multiplicatively) of invertible elements of some particular ring of integers modulo a number hard to factor. The encryption of a message m is always a group element of the form $E(m, r) = g^m r^e \in \mathcal{G}$ where e is some public integer, g some fixed public element in \mathcal{G} , and r is chosen at random in some particular multiplicative subgroup \mathcal{R} of \mathcal{G} . Since \mathcal{R} is a subgroup, such schemes have the desired additive homomorphic property: an encryption of $m_1 + m_2$ can be obtained from any encryption of m_1 and m_2 , as

$$E(m_1, r_1) \cdot E(m_2, r_2) = E(m_1 + m_2, r_1 r_2) .$$

5.2.1 The Early Mechanisms (80's)

This idea was proposed for the first time by Goldwasser and Micali [16] when they introduced the notion of probabilistic encryption. The Goldwasser-Micali scheme is based on quadratic residues: it selects $\mathcal{M} = \mathbb{Z}_2$, $\mathcal{G} = \mathcal{R} = \mathbb{Z}_N^*$ where $N = pq$ is an RSA-type modulus, $e = 2$ and the base g is chosen to be a pseudo-square modulo N . The (semantic) security of the scheme can be proven with respect to the *Quadratic Residuosity Assumption*³.

³A number $x \in \mathbb{Z}_N^*$ is said to be a *quadratic residue* modulo N if there exists another element $y \in \mathbb{Z}_N^*$ such that $x = y^2 \pmod N$. Now let X_N be the subgroup of \mathbb{Z}_N^* of elements having Jacobi symbol equal to 1. See [17] for a definition of the Jacobi symbol. It is possible to prove that every quadratic residue of \mathbb{Z}_N^* belongs to X_N . The *Quadratic Residuosity Assumption* states that, given N without its factorization, it is computationally infeasible to distinguish quadratic residues in \mathbb{Z}_N^* from random elements in X_N .

The above is generalized as follows. Let p be an integer such that $p | \phi(N)$. We say that $x \in \mathbb{Z}_N^*$ is a *p-residue* if there exists another element $y \in \mathbb{Z}_N^*$ such that $x = y^p \pmod N$. The *p-Residuosity Assumption* states that, given N without its factorization, it is computationally infeasible to distinguish random elements from *p-residues*.

The Benaloh-Fischer scheme [4, 10] later improved the very limited bandwidth of the Goldwasser-Micali scheme by using higher-order residues: it uses the same groups $\mathcal{G} = \mathcal{R} = \mathbb{Z}_N^*$ where $N = pq$ is a product of two large primes, but this time $\mathcal{M} = \mathbb{Z}_e$ where e is a small prime number dividing $\phi(N)$ and such that e^2 does not divide $\phi(N)$. Finally g is set as a non e -th residue modulo N . The semantic security is proved under the *Prime Residuosity Assumption*. However, decryption is quite inefficient as it applies a (Baby-Step-Giant-Step enhanced) exhaustive search to retrieve the message, thus implying that e must be very small.

5.2.2 Improved Constructions (90's)

In 1998, Naccache and Stern [18] proposed a variant of the Benaloh-Fischer scheme which allows a higher bandwidth. This is achieved by taking e not as a prime but as a product of small primes e_1, \dots, e_p such that $\phi(N)$ is divisible by e but by none of the e_i^2 's, and g is a non e_i -th residue modulo N for all i . The semantic security is still proved under the *Prime Residuosity Assumption*.

At the same time, Okamoto and Uchiyama [20] proposed a very different improvement of the Benaloh-Fischer scheme by changing the group structure. They selected $\mathcal{G} = \mathcal{R} = \mathbb{Z}_N^*$ with $N = p^2q$ instead of $N = pq$ as in Goldwasser-Micali, Benaloh-Fischer and Naccache-Stern. Moreover they use $\mathcal{M} = \mathbb{Z}_p$, $e = N$ and g such that the order of g^p modulo p^2 is $p - 1$. The semantic security is proved under the *p-Subgroup Assumption*⁴, but there exist very simple chosen-ciphertext attacks that can recover the factorization of N . The scheme reaches an encryption ratio similar to the one of Naccache-Stern, however the decryption is more efficient.

More recently, Paillier proposed in [21] an extension of the Okamoto-Uchiyama encryption scheme where N is a usual RSA-modulus but all the algebraic operations are carried out on the group $\mathbb{Z}_{N^2}^*$. More precisely, $N = pq$, $\mathcal{M} = \mathbb{Z}_N$, $\mathcal{G} = \mathbb{Z}_{N^2}^*$, $\mathcal{R} = \mathbb{Z}_N^*$, $e = N$ and g is an element of order divisible by N . The semantic security is proved under the *Decisional Composite Residuosity Assumption* which stipulates that given $N = pq$, it is computationally hard to decide whether a given element of $\mathbb{Z}_{N^2}^*$ is the N -th power of some other element of $\mathbb{Z}_{N^2}^*$. The resulting system is more efficient than the previously mentioned schemes. Besides, no adaptive chosen-ciphertext attacks recovering the secret key is known. For these reasons, Paillier's cryptosystem is currently the best candidate of an encryption scheme with additive homomorphism.

5.2.3 Extensions of Paillier's Scheme

At PKC'01, Damgård and Jurik [13] presented a generalized (and still homomorphic) version of Paillier's basic cryptosystem in which the expansion factor is reduced and the block length of the scheme may be changed adaptively without altering the public key. Moreover, they show that such a variant is as secure as Paillier's original construction.

More recently Cramer and Shoup [12] proposed a very general and elegant methodology to obtain security against adaptive chosen-ciphertext attacks from a certain class of cryptosys-

⁴The *p-Subgroup Assumption* informally states that given $N = p^2q$ without its factorization, it is infeasible to decide if a random element $z \in \mathbb{Z}_N^*$ has order $p - 1$ in $\mathbb{Z}_{p^2}^*$ or not.

tems with some well-defined algebraic properties. In particular they showed how to modify Paillier’s original scheme in order to achieve such a strong security goal. The resulting variant, moreover, allows for a double decryption mechanism: one can decrypt either if the factorization of the modulus is available or if some specific discrete logarithm is known.

Building on this idea, Bresson *et al.* [7] further investigated the basic Cramer-Shoup variant and show that by slightly modifying the underlying structure of the scheme one can obtain a different scheme that allows for some additional applications (see [7] for details), while maintaining, basically all the “good” properties of the Cramer Shoup variant. Moreover the security of the scheme in [7] is based on a different (non residuosity-related) decisional assumption⁵. As the one in [12], this scheme allows for a double decryption mechanism.

5.3 Cryptographic Applications of Homomorphic Encryption

The term *privacy homomorphism* dates back to the early ages of cryptography and is still used to designate a homomorphic trapdoor function, whatever the definition of the group operation happens to be. Rivest, Adleman, and Dertouzos [22] noted applications of trapdoor homomorphisms to computing on encrypted data soon after the introduction of RSA. Brickell and Yacobi [8] broke a number of candidate constructions of privacy homomorphisms.

Boyar *et al.* showed in [6] that a XOR-homomorphic bit commitment could be exploited to yield more efficient zero-knowledge proofs of circuit satisfiability. Benaloh [4] provided a secure election scheme based on his homomorphic encryption scheme. Cramer and Damgård [11] use homomorphic bit commitments to drastically simplify the design of zero-knowledge proofs. Many other examples exist in which homomorphic properties are used to construct cryptographic protocols.

Homomorphic encryption finds applications in a wide range of cryptographic techniques. Traditionally, electronic voting schemes make use of additive encryption for protecting the privacy of ballots while allowing agents to sum up the votes and evaluate the outcome. Auctions are also a typical application but may rely on multiplicative rather than additive encryption. Homomorphicity is also desirable in various contexts of multi-party computation such as non-interactive two-player secure function evaluation [1].

5.4 Further Research: Algebraic Encryption

An *algebraically* homomorphic encryption scheme defined over a ring \mathcal{M} is a *ring* homomorphism instead of being based on a group structure. Thus both inner laws of the underlying ring $(\mathcal{M}, +, *)$ are supported, meaning that there exist efficient algorithms **plus** and **times** defined over the ciphertext space $\text{Im}(E) = \mathcal{B}$ such that for any $m_1, m_2 \in \mathcal{M}$ and any random r_1, r_2 one has (in infix notation)

$$\begin{aligned} E(m_1, r_1) \text{ plus } E(m_2, r_2) &= E(m_1 + m_2, r) \\ E(m_1, r_1) \text{ times } E(m_2, r_2) &= E(m_1 * m_2, r') \end{aligned}$$

⁵Here, by *non-residuosity related assumption*, we mean a decisional assumption which claims something different from the intractability of deciding membership in a set of a high-degree residues.

for some r, r' . Of course, this definition also comprises deterministic encryption *i.e.* when no randomness is involved. Thus, algebraic encryption extends much further the means to compute on encrypted data, thereby opening new ways towards designing cryptographic protocols that are currently unknown or impractical.

Unfortunately, no such (secure) scheme is known so far, despite several attempts reported in the literature. A cryptosystem proposed by Domingo-Ferrer [14] was later broken by Wagner in [25]. Becker, Rappe and Sander [3] introduced the possibility of constructing algebraic encryption from homomorphic encryption over non-abelian groups. Independently, Boneh and Lipton [5] showed that any deterministic cryptosystem that is a ring homomorphism is subject to a subexponential attack in connection with the so-called *black-box field problem*. They further conjectured that any algebraically homomorphic cryptosystem, also described as *completely malleable*, would prove to be insecure.

Feigenbaum and Merritt noted that a cryptosystem which is a ring homomorphism on $\mathcal{M} = \mathbb{Z}/2\mathbb{Z}$ could be used to implement completely non-interactive secure circuit evaluation [15]. Of particular interest in this regard is Sander, Young and Yung's clever construction of an encryption algorithm that is both AND and XOR-homomorphic [23]. The authors note that their system is the first cryptosystem homomorphic over a semi-group.

Overall, no completely satisfactory solution is known to exist. It seems that the quest for an algebraic encryption scheme first appeared in [15, pp. 6–7] and is considered ever since as one of the most important open problems in cryptography.

Acknowledgments

Contributors from ECRYPT: Dario Catalano, Pascal Paillier.

References

- [1] M. Abadi and J. Feigenbaum. Secure circuit evaluation: a protocol based on hiding information from an oracle. In *Journal of Cryptology*, Vol. 3 no. 2, pages 1–12, 1990.
- [2] M. Ajtai and C. Dwork. A public key cryptosystem with worst-case/average-case equivalence. in *Proc. 29th ACM STOC*, p. 284–293, 1997.
- [3] E. Becker, D. Rappe and T. Sander. Homomorphic Encryption on Non-Abelian Groups and Applications, Preprint, 2002.
- [4] J.C. Benaloh. Verifiable Secret-Ballot Elections. Ph.D. Thesis, Yale University, 1988.
- [5] D. Boneh and R. J. Lipton. Algorithms for black-box fields and their application to cryptography. In *Proceedings of CRYPTO '96*, LNCS vol. 1109, pages 283–297, 1996.
- [6] J. Boyar, I. Damgård and R. Peralta. Short Non-Interactive Cryptographic Proofs. In *Journal of Cryptology*, Vol. 13 no. 4, pages 449–472, Autumn 2000.
- [7] E. Bresson, D. Catalano and D. Pointcheval. A Simple Public Key Cryptosystem with a Double Trapdoor Decryption Mechanism and its Applications. In *Proc. of Asiacrypt'03*, LNCS vol. 12894, pages 37–54, 2003.

- [8] E. F. Brickell and Y. Yacobi. On privacy homomorphisms. In *David Chaum and Wyn L. Price, editors, Advances in Cryptology, EUROCRYPT '87*, Springer-Verlag, LNCS vol. 1304, pages 117–126, 1987.
- [9] B. Chor and R. Rivest. A knapsack-type cryptosystem based on arithmetic in finite fields. In *IEEE Transactions on Information Theory*, vol. IT-34 pp.901–909, 1988.
- [10] J. D. Cohen and M. Fischer. A robust and verifiable cryptographically secure election scheme. In *Proc. of the 26th FOCS*. IEEE, 1985.
- [11] R. Cramer and I. Damgård. Zero knowledge proofs for Finite Field arithmetic or Can Zero Knowledge Be For Free? In *Advances in Cryptology, CRYPTO '98*, Springer-Verlag, 1998.
- [12] R. Cramer and V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Eurocrypt '02*, LNCS vol. 12332, pages 45–64, Springer-Verlag, 2002.
- [13] I. Damgård and M. Jurik. A generalization, a simplification and some applications of Paillier's probabilistic public-key system. In *Proc. of PKC '2001*, LNCS vol. 1992, Springer-Verlag, 2001.
- [14] J. Domingo-Ferrer. A Provably Secure Additive and Multiplicative Privacy Homomorphism. In *ISC 2002*, pages 471–483, 2002.
- [15] J. Feigenbaum and M. Merritt. Open questions, talk abstracts, and summary of discussions. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–45, 1991.
- [16] S. Goldwasser and S. Micali. Probabilistic Encryption. *JCSS*, 28(2):270–299, April 1984.
- [17] N. Koblitz *A Course in Number Theory And Cryptography*, 2nd Edition, Springer Verlag.
- [18] D. Naccache and J. Stern. A New Public Key Cryptosystem Based on Higher Residues. In *Proc. of 5th Symposium on Computer and Communications Security*. ACM, 1998.
- [19] P.Q. Nguyen and J. Stern. Cryptanalysis of the Ajtai-Dwork Cryptosystem. In *Advances in Cryptology - Crypto '98*, LNCS vol. 1462, Springer Verlag, 1999, pages 223-242.
- [20] T. Okamoto and S. Uchiyama. A New Public-Key Cryptosystem as Secure as Factoring. In *Advances in Cryptology - Eurocrypt '97*, LNCS vol. 1233, Springer, 1997, pages 308-318.
- [21] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology - Eurocrypt '99*, LNCS vol. 1592, Springer Verlag, 1999, pages 223-238.
- [22] R. Rivest, L. Adleman and M. Dertouzos. On Databanks and Privacy Homomorphisms. In *Foundations of Secure Computation*, R. A. DeMillo et al., Editors, Academic Press, Inc., New York, 1978, pages 168-177.

- [23] T. Sander, A. Young and M. Yung. Non-interactive cryptocomputing in NC1. In *FOCS '99*, 1999.
- [24] S. Vaudenay. Cryptanalysis of the Chor-Rivest Cryptosystem. In *Advances in Cryptology - Crypto '98*, LNCS vol. 1462, Springer Verlag, 1999, pages 243-256.
- [25] D. Wagner. Cryptanalysis of an Algebraic Privacy Homomorphism. In *ISC 2003*, 2003.

6 Identity-based cryptography

6.1 Description and Motivations

The concept of identity-based cryptography has been put forth by A. Shamir [Sha84]. In identity-based public-key encryption schemes, the problem of distributing the public keys is avoided by making the public key derivable from some known aspect of the user identity (for example, the email address). For obtaining the secret keys, instead, identity-based encryption requires the existence of a trusted key generator (TKG) that, after having authenticated a user, releases the secret key associated with the identity of the user.

The main advantage of identity-based is that it dispenses with the need of an authenticated public-key directory. This can be replaced by a directory containing the public parameters of the trusted party. It is expected that the number of TKG's is substantially smaller than the number of user making the maintenance of this directory less burdensome.

The following are the drawbacks of identity-based encryption:

1. a TKG knows the secret keys of all the users;
2. each user has to authenticate to the TKG just as he would in the traditional public-key model;
3. a secure channel is needed to transfer the secret of a user from the TKG to the user;
4. a user has to publish the public parameters of his TKG so that other users can send him messages.

6.1.1 Notions of security for identity based encryption

Chosen ciphertext security is one of the standard notions of security for cryptosystems. For identity-based encryption we modify this notion since an adversary might have the secret keys relative to identities of his choice. Thus we consider the following game for defining the security of identity based encryption schemes with respect to *chosen ciphertext attacks*.

The TKG publishes the public parameters for the system and keeps secret the private master key. The adversary can issue both extraction queries (ID) (in which he asks for the secret key of an identity ID of his choice) or decryption keys (c, ID) (in which he asks for the decryption of a ciphertext c with respect to an identity ID of his choice). Then the adversary outputs an identity ID^* (on which he had not asked for the corresponding private key) and two messages

m_0 and m_1 . Then b is chosen at random and an encryption c^* of m_b with respect to identity ID is computed and given to the adversary. The adversary, before giving his guess b' for b , is allowed to ask extraction queries for identities $ID \neq ID^*$ and decryption queries for $(c, ID) \neq (c^*, ID^*)$. We say that the adversary breaks the system if $b = b'$ with probability significantly better than $1/2$.

Selective identity security is a weaker security model for identity-based encryption. In this model the adversary must commit ahead of time to the identity it intends to attack, whereas in the standard model the adversary is allowed to choose the identity adaptively. It is easy to show that any selective identity secure encryption scheme can be converted to a fully secure scheme by restricting the space of identities, but the proof uses an inefficient security reduction [BB04b].

6.1.2 Identity-based identification and signatures

A *standard* identification (SI) scheme is a triplet of algorithms $SI = (\text{Kg}, \text{P}, \text{V})$. The prover generates a key pair (pk, sk) using the key generation algorithm Kg . He publishes the public key pk and keeps the secret key sk secret. The prover can then prove his identity to a verifier by running the prover algorithm P initiated on sk in interaction with the verification algorithm V run by the verifier on input pk . Security requires that no polynomial-time adversary has a reasonable probability of making the verifier accept after having seen the public key and either a number of valid conversation transcripts (passive attack), or after interacting with a real prover sequentially (active attack) or arbitrarily (concurrent attack).

Analogously, an *identity-based identification* (IBI) scheme $IBI = (\text{MKg}, \text{UKg}, \bar{\text{P}}, \bar{\text{V}})$ is defined as follows. The trusted key generation center runs the master key generation algorithm MKg to generate a master key pair (mpk, msk) . The master public key mpk is made available as a system-wide parameter, the master secret key msk is kept secret. When a user with identity I registers at the key distribution center, the center uses the user key generation algorithm UKg on input msk, I to generate the user secret key usk corresponding to identity I , and sends usk to the user over a secure and authenticated channel. To prove his identity, the prover runs algorithm $\bar{\text{P}}$ on input usk and interacts with the verifier who runs the $\bar{\text{V}}$ algorithm on input mpk, I . Note that the verification algorithm needs no user-specific information about the prover other than his identity. For an *identity-based signature* (IBS) scheme $IBS = (\text{MKg}, \text{UKg}, \overline{\text{Sign}}, \overline{\text{Vf}})$, the MKg and UKg algorithms are defined in the same way. The signer computes a signature for a message M by running the $\overline{\text{Sign}}$ algorithm on input his user secret key usk and M . This signature can later be verified through the verification algorithm based on the master public key and the identity of the user.

6.2 State of the art

6.2.1 Pairing

Some of the constructions of id-based encryption are based on the concept of bilinear mapping (or pairing) that we briefly review.

Let G_1 and G_2 be two groups of prime order q . We view G_1 as an additive group and G_2 as a

multiplicative group. We assume that the discrete logarithm problem is hard in both groups. A mapping $e : G_1 \times G_1 \rightarrow G_2$ satisfying the following properties is called a *pairing*:

1. Bilinearity. For all $P, Q \in G_1$ and $R, S \in G_2$,

$$e(P + Q, R) = e(P, R)e(Q, R)$$

and

$$e(P, R + S) = e(P, R)e(P, S).$$

2. Non-degeneracy. For all generators P of G_1 , $e(P, P)$ is a generator of G_2 .

We also assume that $e(\cdot, \cdot)$ is computable in polynomial time. Examples of pairings are the Weil pairing (see [BF03]) and the Tate pairing (see [GHS02]).

6.2.2 Identity Based Encryption with Random Oracle

Identity Based Encryption based on Pairing In this section we describe the Identity Based encryption scheme of [BF03].

We describe protocols **Setup**, used by the TKG to generate the public parameters and the private key, **Extract**, used by the TKG to generate the secret key corresponding to an identity, **E**, used to encrypt a message, and **D**, used to decrypt a message.

We assume that groups G_1, G_2 , generator P for G_1 and pairing e are known to all users. Moreover we denote by $H_1 : \{0, 1\}^* \rightarrow G_1$ and by $H_2 : G_2 \rightarrow \{0, 1\}^n$ two hash functions.

1. Algorithm **Setup**.

Choose s at random from Z_q^* . Public key is $pk = (sP)$ and secret key s .

2. Algorithm **Extract**.

The secret key sk_{ID} associated to identity ID is $sk_{ID} = sH_1(ID)$.

3. Algorithm **E**.

To encrypt message M , choose r at random from Z_q^* and output ciphertext C consisting of the pair

$$C = (rP, M \oplus H_2(e(H_1(ID), pk)^r)).$$

4. Algorithm **D**.

To decrypt ciphertext $C = (U, V)$, compute

$$V \oplus H_2(e(sk_{ID}, U)).$$

Efficiency. Each algorithm requires at most two application of the hash function. Moreover, the **Setup** and the **Extract** algorithm require only one scalar multiplication in G_1 . Encryption is computed using one scalar multiplication, one pairing and one group exponent. Decryption only requires one pairing computation.

Security. The encryption scheme above is secure with respect to *chosen ciphertext attacks* in the random oracle model under the assumption that the following computational problem is hard.

Assumption 6.1 (Bilinear Diffie-Hellman problem (BDH problem)) *Fix groups G_1 and G_2 and a pairing e . An instance of the BDH problem consists of a generator P of G_1 and elements $A = aP$, $B = bP$ and $C = cP$ of G_1 . The problem consists in computing $e(P, P)^{abc}$.*

Identity Based Encryption based on Quadratic Residuosity In this section we describe a scheme by Cocks [Coc01]. The scheme assumes a hash function H that maps IDs to elements of Z_n^* with Jacobi symbol $+1$.

1. Algorithm Setup.

Generate a random Blum integer $n = p \cdot q$. The public key of the TKG is n and the secret is (p, q) .

2. Algorithm Extract.

The secret key sk_{ID} associated with identity ID is the square root modulo n of $H(\text{ID})$ or of $n - H(\text{ID})$ (notice that since n is a Blum integer at least one of $H(\text{ID})$ and $-H(\text{ID})$ is a perfect square modulo n).

3. Algorithm E.

The encryption of a one-bit message $b \in \{+1, -1\}$ with respect to identity ID is computed by picking t, t' at random from Z_n^* with Jacobi symbol equal to b and computing

$$c_1 \leftarrow t + \frac{H(\text{ID})}{t}$$

$$c_2 \leftarrow t' + \frac{-H(\text{ID})}{t'}.$$

The ciphertext c is finally set equal to $c = (c_1, c_2)$.

4. Algorithm D.

The cleartext b corresponding to ciphertext $c = (c_1, c_2)$ is computed in the following way

$$b = \begin{cases} \left(\frac{2sk_{\text{ID}} + c_1}{n} \right), & \text{if } H(\text{ID}) \text{ is a square modulo } n; \\ \left(\frac{2sk_{\text{ID}} + c_2}{n} \right), & \text{if } -H(\text{ID}) \text{ is a square modulo } n. \end{cases}$$

6.2.3 Identity Based Encryption without Random Oracle

In this section we describe a scheme by Boneh and Boyen [BB04a]. The scheme is only selective identity secure. ID's are assumed to be elements of Z_q^* and messages are elements of G_2 .

1. Algorithm Setup.

Randomly pick $x, y \in Z_q^*$ and set $U = xP$ and $V = yP$. The public key of the TKG is $pk = (U, V)$ and the secret key is (x, y) .

2. Algorithm Extract.

The secret key $sk_{\text{ID}} = (r, K)$ corresponding to ID is computed by picking $r \in Z_q^*$ and by setting $K = P/(\text{ID} + x + ry)$.

3. Algorithm E.

To encrypt message $M \in G_1$ under public key ID compute ciphertext

$$C = (\text{ID} \cdot s \cdot P + s \cdot U, s \cdot V, e(P, P)^s \cdot M),$$

where s is randomly chosen from Z_q^* .

4. Algorithm D.

To decrypt ciphertext $C = (X, Y, Z)$ using private key $sk_{\text{ID}} = (r, K)$ output message $M = Z/e(X + rY, K)$.

Security. The encryption scheme above is secure with respect to selective-ID chosen ciphertext attack under the assumption that the following computational problem is hard.

Assumption 6.2 (Bilinear Diffie-Hellman Inversion) Given the tuple $(P, xP, x^2P, \dots, x^qP)$ compute $e(P, P)^{1/x}$.

Efficiency considerations. As remarked above, a selective-ID secure identity based encryption scheme can be converted into a fully secure scheme (see [BB04b]). However the reduction is not security preserving. For a concrete example, suppose that identities in the system are 160-bit long and that we use an encryption scheme such that no t -time-bounded adversary has advantage at most 2^{-240} in a selective ID attack. Then the reduction of [BB04b] constructs a system in which the adversary in a fully adaptive attack has advantage at most 2^{-80} (up from 2^{-240} of the original scheme). In [BB04b], Boneh and Boyen give a fully secure identity based encryption scheme whose security is established via a polynomial reduction from the decisional bilinear Diffie-Hellman assumption (the security does not assume a random oracle). The scheme presented is impractical and a more efficient construction is presented in [Wat05].

6.2.4 Hierarchical Identity Based Encryption

Hierarchical Identity Based Encryption is a generalization of identity based encryption in which identities are organized in a hierarchy tree. An identity can issue private keys to all of its descendant identities but cannot decrypt messages intended for identities that are not its descendant.

The first construction in the random oracle model has been given in [GS02] and a construction without the random oracle has been presented in [BB04a]. In both constructions the length

of the ciphertext and private keys grows linearly with the level of the identity. In [BBG05] a construction is presented in which ciphertext length and decryption costs are independent of the level of the identity. The security does not assume the existence of a random oracle and is based on the bilinear Diffie-Hellman exponent assumption.

Currently, there are two main applications for hierarchical identity based encryption. The first one is to construct forward secure encryptions [CHK03] and the second one is to construct public-key broadcast systems.

6.2.5 Fuzzy Identity Based Encryption

In a *fuzzy* identity based encryption [SW05], identities are seen as sets of descriptive attributes. A user with the secret key for identity ID is able to decrypt a ciphertext encrypted with the public key ID' if and only if ID and ID' are “close enough.” This new kind of encryption can be used in a scenario in which the identity of a user is related to some biometrics (for example iris scan) and, since biometric measurements are noisy, some level of flexibility is needed. Also fuzzy identity-based encryptions can be used to construct “attribute-based encryption” in which identities are lists of attributes and a user can decrypt all messages encrypted with identities that are contained in his.

In [SW05], a construction for fuzzy identity based encryption is presented and it is shown that, under an assumption similar to the decisional bilinear Diffie-Hellman assumption, the construction enjoys a modified version of selective ID security. The construction does not require random oracles.

6.2.6 Identity Based Identification and Signature

The late eighties and early nineties saw the proposal of many IBI and IBS schemes. These include the IBS scheme in Shamir's paper [Sha84] that introduced the concept of identity-based cryptography, the Fiat-Shamir IBI and IBS schemes [FS86], the Guillou-Quisquater IBI and IBS schemes [GQ89], and others [Oka93, Gir90, Bet88]. More recently, new pairing-based IBS schemes have been proposed [SOK00, Hes03, Pat02, CC03, Yi03].

Although there is a lot of work on proving security for identification and signature schemes, for a long time it pertained 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, it lasted until the independent work of Kurosawa and Heng [KH04] and of Bellare, Namprempre and Neven [BNN04] for a security definition to be proposed taking into account the identity-based aspects of IBI schemes.⁶ Cha and Cheon provide a definition of security for IBS schemes and prove their own scheme secure [CC03].

Security proofs through transformations. Dodis, Katz, Xu, and Yung [DKXY03] define a class of standard signature (SS) schemes that they call trapdoor, and then present

⁶The notion of [KH04] is somewhat weaker than that of [BNN04] because it does not allow the adversary to corrupt identities during the impersonation.

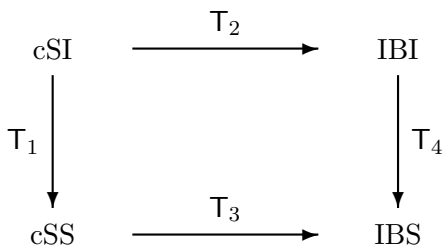


Figure 2: Family of schemes associated to a cSI scheme.

a random-oracle-using transformation that turns any secure trapdoor SS 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 the transform of [DKXY03] to underlying trapdoor SS schemes already proven secure in the literature [PS00, OO98, AABN02].

Kurosawa and Heng [KH04] show how to construct an IBI scheme from any SS scheme using honest-verifier zero-knowledge proofs of knowledge of a signature. The resulting IBI scheme, however, is only secure against passive attacks. Bellare et al. [BNN04] define a class of *convertible* SI (cSI) schemes, and present an entire framework of security-preserving (in the random oracle model) transformations converting cSI schemes into SS, IBI and IBS schemes. Thus, they view schemes as occurring in families, as depicted in Figure 2. The first transform T_1 from a cSI scheme into a (convertible) SS scheme is the well-known Fiat-Shamir transform [FS86]. The resulting cSS scheme is known to be unforgeable under chosen-message attack if the cSI scheme is secure against impersonation under passive attack [AABN02]. The second transform T_2 converts a cSI scheme, that is secure against impersonation under passive, active or concurrent attack, into an IBI scheme that is secure under the same type of attack. Transform T_3 is a generalization of the transform of [DKXY03] and converts a cSS scheme into an IBS scheme while preserving the unforgeability under chosen-message attack. Finally, transform T_4 is the analogue of T_1 for the identity-based setting, converting IBI schemes into IBS schemes. The diagram commutes in the sense that $T_3(T_1(\text{cSI})) = T_4(T_2(\text{cSI}))$. Hence, proving the security of the resulting IBS scheme is reduced to proving the original cSI scheme secure against impersonation under passive attack, which in most cases is a considerably simpler task.

The class of cSI schemes seems wide enough to capture most concrete schemes appearing in the literature. In particular, [BNN04] found the schemes of [Sha84, FS86, Bet88, FFS88, GQ89, Gir90, OO90, OS90, Oka93, SOK00, FF02, Hes03, CC03, Yi03] all to originate from a cSI scheme – even if this cSI scheme had not been defined in the literature as such – and they provide proofs (or in the case of [Gir90], attacks) for them. They also directly prove the security of two exceptional IBI schemes that escape being captured by their framework.

6.3 Open problems

Filling the gaps left by [BNN04]. In spite of the vast amount of schemes covered by [BNN04], a few questions remain unanswered. The first concerns the security of the so-called iterated-root scheme under concurrent attack. A second problem is the active and concurrent

security of the scheme of [Bet88], and also the security of the more general formulation of the same scheme with longer public keys under any attack. Lastly, a scheme by Paterson [Pat02] seems to resist attempts to both attack and proof.

Tightness of reductions. The quadrangle of transformations in Figure 2 only considers the first-order concern of proving the asymptotical security of IBI and IBS schemes. A second-order concern is the exact security of the reductions. Due to the generality of the transformations in Figure 2, the reductions are rather loose. For example, according to the proof of [BNN04], it takes a key length of 19611 bits to make breaking Shamir’s IBS scheme as hard as breaking the one-wayness of RSA with 1024 bits! For some IBI and IBS schemes, tighter proofs can be obtained by proving them directly, rather than through the transforms, and exploiting algebraic properties of the scheme. Interestingly, Libert and Quisquater [LQ04] found that a variant of the IBS scheme of [SOK00] has a tight security reduction from the computational Diffie-Hellman assumption.

Acknowledgments

Contributors from ECRYPT: Clemente Galdi, Gregory Neven, Giuseppe Persiano, Ivan Visconti.

References

- [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, Berlin Germany, April 2002.
- [BB04a] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology: Proceedings of EUROCRYPT 04*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 2004.
- [BB04b] Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology: Proceedings of CRYPTO 04*, volume 3152 of *Lecture Notes in Computer Science*, pages 443–459. Springer-Verlag, 2004. See also extended version available from the authors’ homepages.
- [BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology: Proceedings of EUROCRYPT 05*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer-Verlag, 2005.
- [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, Berlin Germany, May 1988.
- [BF03] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil Pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.
- [BNN04] 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*, pages 268–286. Springer-Verlag, Berlin Germany, 2004.
- [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, Berlin Germany, August 2002.
- [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, Berlin Germany, January 2003.
- [CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *Advances in Cryptology: Proceedings of EUROCRYPT 03*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer-Verlag, 2003.
- [Coc01] Clifford Cocks. An indentity based encryption scheme based on quadratic residuosity. In *IMA International Conference on Cryptography and Coding*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–263. Springer-Verlag, 2001.
- [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, Berlin Germany, January 2003.
- [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, Berlin Germany, February 2002.
- [FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.
- [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, Berlin Germany, August 1986.

- [GHS02] S. Galbraith, K. Harrison, and D. Soldera. Implementing Tate Pairing. In *Algorithmic Number Theory Symposium*, volume 2369 of *Lecture Notes in Computer Science*, pages 324–337. Springer-Verlag, 2002.
- [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, Berlin Germany, May 1990.
- [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, Berlin Germany, 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, Berlin Germany, 2002.
- [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.
- [KH04] Kaoru Kurosawa and Swee-Huay Heng. From digital signature to ID-based identification/signature. In F. Bao, R. Deng, and J. Zhou, editors, *Advances in Cryptology – Public-Key Cryptography 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag, Berlin Germany, 2004.
- [LQ04] Benoît Libert and Jean-Jacques Quisquater. The exact security of an identity based signature and its applications. Cryptology ePrint Archive, Report 2004/102, 2004. <http://eprint.iacr.org/>.
- [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, Berlin Germany, 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, Berlin Germany, 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, Berlin Germany, 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, Berlin Germany, May 1990.

- [Pat02] Kenneth G. Paterson. ID-based signatures from pairings on elliptic curves. Technical Report 2002/004, IACR ePrint Archive, January 2002.
- [PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
- [Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology: Proceedings of CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1985, 1984.
- [SOK00] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *SCIS 2000*, Okinawa, Japan, January 2000.
- [SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology: Proceedings of EUROCRYPT 05*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer-Verlag, 2005.
- [Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology: Proceedings of EUROCRYPT 05*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer-Verlag, 2005.
- [Yi03] Xun Yi. An identity-based signature scheme from the Weil pairing. *IEEE Communications Letters*, 7(2):76–78, 2003.