

STORK

strategic roadmap for crypto

Project Number	IST-2002-38273
Project Title	STORK
Deliverable Type	Report
Security Class	Public
Deliverable Number	D4
Title of Deliverable	New Trends in Cryptology
Document Reference	ENS-D4-1.4
Editor	Phong Nguyen (CNRS/ENS)
Abstract	This is the document identifying the problems faced by cryptographers and users of cryptology, either currently or in the short or medium term future.
Keywords	STORK, new trends, cryptology.

Version 1.4

May 22, 2003

New Trends in Cryptology[†]

B. Preneel¹, A. Bosselaers¹, A. Biryukov¹,
J. Stern², D. Catalano², L. Granboulan², P. Nguyen², D. Pointcheval²,
H. Dobbertin³, T. Lange³, P. Felke³, C. Paar³,
H. Handschuh⁴,
K. Nguyen⁵, P. P. Roelse⁶,
S. Babbage⁷,
M. Näslund⁸,
H. Gilbert⁹.

May 22, 2003
Version 1.4

[†]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-2002-38273. 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.

¹Katholieke Universiteit Leuven, Dept. Elektrotechniek-ESAT/COSIC, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium, {bart.preneel, antoon.bosselaers, alex.biryukov}@esat.kuleuven.ac.be

²École Normale Supérieure, Département d'Informatique, 45 rue d'Ulm, Paris 75230 Cedex 05, France, {jacques.stern, louis.granboulan, phong.nguyen, david.pointcheval}@ens.fr

³Ruhr-University Bochum, Horst-Görtz-Institute for IT-Security, NA 5/73, Universitätsstraße 150, D-44780 Bochum, Germany, {hans.dobbertin@, lange@itsc., patrick.felke@, cpaar@crypto.}ruhr-uni-bochum.de

⁴Gemplus S.A., R&D Security Technologies Department, 34 rue Guynemer, Issy-les-Moulineaux 92447, France, helena.handschuh@gemplus.com

⁵Philips Semiconductors GmbH, Cryptology Competence Center, PO Box 540 240, 22502 Hamburg, Germany, kim.nguyen_3@philips.com

⁶Philips Semiconductors, Cryptology Competence Center Prof. Holstlaan 4, NL-5656 AA Eindhoven, The Netherlands, peter.roelse@philips.com

⁷Vodafone Group R&D, Vodafone House, The Connection, Newbury, Berkshire RG14 2FN, UK, steve.babbage@vodafone.com

⁸Ericsson Research, Communications Security Lab, Torshamnsgatan 23, 16480 Stockholm, Sweden, mats.naslund@era-t.ericsson.se

⁹France Télécom R&D, 38-40 rue du General Leclerc, Issy-les-Moulineaux 92794, France, henri.gilbert@rd.francetelecom.fr

Contents

1	Introduction	1
2	Cryptology in the Information Technology Society	3
2.1	Secure communication	3
2.1.1	Mobile system security	3
2.1.2	Network security	5
2.2	Financial services	6
2.2.1	Digital wallet, electronic cash and micro-payments	6
2.3	Social aspects	8
2.3.1	Ambient intelligence	8
2.3.2	Timestamping	10
2.3.3	E-voting	13
2.3.4	Digital Rights Management	16
3	Cryptographic protocols	19
3.1	Two party protocols	19
3.1.1	On-line authentication (identification)	19
3.1.2	Off-line authentication (of a document)	20
3.1.3	Secure communication channel	20
3.2	Multiparty protocols	21
3.2.1	Multiparty computation	21
3.2.2	Formal Models of Security and Composability of Protocols	22
3.2.3	Key management and key establishment	23
3.2.4	Public Key Infrastructures	24
3.2.5	Privacy and anonymity	25
3.2.6	Quantum Cryptographic Protocols	27
4	Cryptographic techniques	28
4.1	Symmetric cryptography	28
4.1.1	Pseudo-random permutations: block ciphers	28
4.1.2	Stream ciphers	31
4.1.3	Pseudo-random number generation	33

4.1.4	Pseudo-random functions	36
4.1.5	Cryptographic hash functions	37
4.1.6	Message Authentication Codes	38
4.2	Asymmetric cryptography	38
4.2.1	Public key encryption	38
4.2.2	Digital signature	42
4.3	Implementation aspects and realization	46
4.3.1	Flexibility	46
4.3.2	Efficient hardware	47
4.3.3	Efficient software	48
4.3.4	Embedded software	48
4.4	Dedicated attack techniques	50
4.4.1	Side-channel attack of an implementation	50
4.4.2	Active Attacks on an Implementation: Fault Induction	52
4.4.3	Alternative computing devices	53
5	Mathematical foundations	58
5.1	Theory of computation	58
5.1.1	Complexity theory	58
5.1.2	Information theory	61
5.2	Combinatorics in Finite Fields	65
5.3	Algorithmic number theory	66
5.3.1	Integer factorization	66
5.3.2	Discrete logarithm in finite fields	66
5.3.3	Algebraic curves over finite fields	67
5.3.4	Geometry of numbers	69
5.3.5	Systems of multivariate polynomial equations	69
5.4	Combinatorial Group Theory	70

1 Introduction

Cryptographic algorithms play a crucial role in the information society. When we use our credit card or an ATM, call someone on a mobile phone, get access to health care services, watch pay-per-view channels, or buy something on the web, cryptographic algorithms are used to offer protection. These algorithms guarantee that nobody can steal money from our account, place a call at our expense, eavesdrop on our phone calls, or get unauthorized access to sensitive health data. It is clear that information technology will become increasingly pervasive: in the short term we expect to see more of e-government, e-voting, m-commerce, . . . ; beyond that we can expect the emergence of ubiquitous (or pervasive) computing, ambient intelligence, . . . These new environments and applications will present new security challenges, and there is no doubt that cryptographic algorithms and protocols will form part of the solution.

While cryptography is an essential component, the importance of cryptography should be put in the correct perspective. Indeed, failure of security systems can often be blamed on other reasons than failure of cryptography. (see for example Anderson [1]). Nevertheless, cryptographic algorithms are part of the foundations of the security house, and any house with weak foundations will collapse. There is thus no excuse whatsoever to employ weak cryptography; nevertheless, we encounter weak cryptography more frequently than necessary and this for several reasons:

- Cryptography is a fascinating discipline, which tends to attract ‘do-it-yourself’ people, who are not aware of the scientific developments of the last 25 years; their home-made algorithms can typically be broken in a few minutes by an expert;
- Use of short key lengths, in part due to export controls (mainly in the US, who dominates the software market) which limited key sizes to 40 bits (or 56 bits) for symmetric ciphers, 512 bits for factoring based systems (RSA) and discrete logarithm modulo a large prime (Diffie-Hellman). The US export restrictions have been lifted to a large extent in January and October 2000 (see Koops [4] for details). In several countries, domestic controls were imposed; the best known example is France, where the domestic controls were lifted in January 1999. Nevertheless, it can take a long time before all applications are upgraded.
- Progress in cryptanalysis: open academic research has started in the mid 1970ies; cryptology is now an established academic research discipline, and the IACR (International Association for Cryptologic Research) has more than 1000 members. As a consequence of this, increasingly sophisticated techniques are developed to break cryptosystems, but fortunately also to improve their security.
- Progress in computational power: Moore’s law, which was formulated in 1965, predicts that every 18 months transistor density will double. Empirical observations have proved him right (at least for data density) and experts believe that this law will be holding for at least another 15 years. The variation of Moore’s law for computational power states that the amount of computation that can be done for the same cost doubles every 18 month. This implies that a key for a symmetric algorithm will become thousand times cheaper to find after 15 years (or needs to increase in length by 10 bits to offer the same

security). An even larger threat may be the emergence of new computer models: if quantum computers can be built, factoring may be very easy (a result by Shor of 1994 [5]). While early experiments are promising [6], experts are divided on the question whether sufficiently powerful quantum computers can be built in the next 15 years. For symmetric cryptography, quantum computers are less of a threat: they can reduce the time to search a $2n$ -bit key to the time to search an n -bit key (using Grover's algorithm [3]). Hence doubling the key length offers an adequate protection.

As a consequence of all these observations, insecure cryptographic algorithms are much more common than they should be. In order to avoid these problems, adequate control mechanisms should be established at several levels:

- Substantial evaluation is necessary before an algorithm can be used; experts seem to agree that a period of 3 to 5 years is required between first publication and use of an algorithm.
- Continuous monitoring is required during the use of a primitive, to verify whether they are still adequate. Especially for public key primitives, which are parameterizable, a rigorous monitoring procedure is required to establish minimal key lengths.
- Adequate procedures should be foreseen to take an algorithm out of service or to upgrade an algorithm. Single DES is a typical example of an algorithm which has been used beyond its lifetime (for most applications, 56 bits was no longer an adequate key length in the 1990ies); another example is the GSM encryption algorithm A5/1: experts agree that it is not as secure as believed (see e.g. Biryukov *et al.* [2]), but it is very difficult to upgrade it.

Especially the last problem should not be underestimated: for data authentication purposes, a new security weaknesses that is discovered will typically not influence older events, and long-term security can be achieved by techniques such as re-signing. However, for confidentiality the problem is much more dramatic: one cannot prevent that an opponent has access to ciphertext, and in certain cases (e.g., medical applications) secrecy for 50–100 years is required. This means that an encryption algorithm used now will need to withstand attacks employed in 2075. It is probably easier to imagine how hard it must have been to design in 1925 an encryption system that needed to be secure for 75 years. There is no reason to believe that this problem is easier at the beginning of the 21st century.

The present document is an attempt to give a state-of-the-art in cryptology, outlining new trends. It is organized as follows. In Section 2, we review the main applications of cryptology in the information technology society. In Section 3, we review the area of cryptographic protocols. In Section 4, we review the basic cryptographic techniques, including encryption, signature and hashing. Eventually, in Section 5, we review the mathematical foundations of cryptology.

Acknowledgments. We are thankful for the comments we received after the presentation of the documents at the first STORK workshop, November 26-27, 2002, in Bruges. Comments on this present version are also warmly welcomed. Please use the STORK discussion forum at

<http://www.stork.eu.org/> or send an email to stork-info@stork.eu.org or to the editor at Phong.Nguyen@ens.fr.

We would like to thank several external researchers who helped us to write the present document. They are, in alphabetical order: Thomas Beth, Christian Cachin, Jan Camenisch, Pierrick Gaudry, Willi Geiselmann, Thomas Johansson, Willi Meier, Joern Mueller-Quade, Stefaan Seys, Igor Shparlinski, Rainer Steinwandt, Pim Tuyls and Karel Wouters.

References

- [1] R.J. Anderson, “Why cryptosystems fail,” *Communications ACM*, Vol. 37, No. 11, November 1994, pp. 32–40.
- [2] A. Biryukov, A. Shamir, D. Wagner, “Real time cryptanalysis of A5/1 on a PC,” *Fast Software Encryption, LNCS 1978*, B. Schneier, Ed., Springer-Verlag, 2000, pp. 1–18.
- [3] L.K. Grover, “A fast quantum mechanical algorithm for database search,” *Proc. 28th Annual ACM Symposium on Theory of Computing*, 1996, pp. 212–219.
- [4] B.-J. Koops, “Crypto law survey,” <http://rechten.kub.nl/koops/cryptolaw>.
- [5] P.W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” *Proc. 35th Annual Symposium on Foundations of Computer Science*, S. Goldwasser, Ed., IEEE Computer Society Press, 1994, pp. 124–134.
- [6] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, I.L. Chuang, “Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance,” *Nature*, 414, 2001, pp. 883–887.

2 Cryptology in the Information Technology Society

2.1 Secure communication

2.1.1 Mobile system security

Mobile systems security does not, in general, drive ground-breaking research in the field of cryptography; it is rather an area of application for existing cryptographic theory. However, there are a number of distinctive aspects of mobile telephony that deserve special mention:

- Since mobile telephone communication is over a radio interface, it is pretty well essential to use stream ciphers rather than block ciphers for encryption. (Single bit errors would have an “avalanche” effect during decryption with block ciphers, and extensive error-correcting techniques are not an option due to bandwidth constraints.)
- Since mobile telephones run on batteries, it is very important that any cryptographic algorithms can be implemented to run on low power. Typically this means hardware implementations (ASICs) with a small number of gates. It also means that algorithms with large initialisation (key setup) overheads are unsuitable.

- The SIM card [3] (a smart card) plays a fundamental role in GSM, UMTS and some other systems. Cryptographic algorithms running on the smart card have to be suitable for these low end platforms.
- It should be difficult for even the owner of a SIM card to extract the cryptographic keys from it. This places strong requirements not just on the algorithms that use those keys, but also on the ways in which the algorithms are implemented. Naïve implementations of cryptographic algorithms on smart cards have been shown to be vulnerable to "side channel attacks" - where the attacker with the smart card in his possession observes not just the digital inputs and outputs of the algorithm, but also other information such as how long the card takes to perform certain operations, how much power it consumes, how it behaves under certain fault-inducing conditions, and so on. These attacks can be very powerful, extracting keys in seconds. However, smart card manufacturers have developed strong implementation techniques to resist such attacks, and at present it seems that side channel attacks make no impression on the most advanced implementations. But this is a fairly young science, and there will probably be further advances on both the attack and defense sides within the period of STORK's influence.
- Special purpose protocols have been developed to secure mobile telephone networks and links. While these cannot generally be said to advance the theory of cryptographic protocols, nevertheless they are an important and high profile instance of protocol selection.
- Again, cryptographic algorithms have been developed especially for use in mobile telephone systems. The block cipher KASUMI [1, 2], and novel constructions of both a stream cipher and a message authentication code from KASUMI, are good recent examples from UMTS. The stream cipher and message authentication code constructions in particular show how improvements can be made over more standard constructions when particular aspects of the context are taken into account.
- The mobile phone (and its SIM) has, thanks to the high level of market penetration, an excellent chance of being the first universal platform for implementing various "personal security services", e.g. performing electronic payments, managing credentials for universal user authentication, etc.
- Mobile terminals are likely to become the first devices that implement an open, standardized Digital Rights Management system.

Note: another collaborative research project PAMPAS (Pioneering Advanced Mobile Privacy And Security) is running in parallel with STORK, and may feed novel requirements for cryptologic research into the STORK process. Close links are being maintained with PAMPAS.

References

- [1] 3GPP TS 35.201. Specification of the 3GPP confidentiality and integrity algorithms; Document 1: f8 and f9 specifications.

- [2] 3GPP TS 35.202. Specification of the 3GPP confidentiality and integrity algorithms; Document 2: Kasumi algorithm specification.
- [3] 3GPP TS 35.206. 3G Security; Specification of the MILENAGE algorithm set: An example algorithm Set for the 3GPP Authentication and Key Generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm specification.

2.1.2 Network security

Network security is, as perhaps can be seen from the name, not core cryptography, but rather one of the most important applications thereof.

As such, the network can be either physical or logical and there is not necessarily a one-to-one correspondence between physical hardware and network. The main security problems are to

- ensure that unauthorized parties (complete outsiders or ones belonging to other logical networks) can not eavesdrop or modify traffic, i.e. to provide intra- and inter-network protection,
- to protect the logical and physical perimeter of the network from outsiders.

The last item usually means special purpose hardware and software such as firewalls, intrusion detection systems etc, and will not be further elaborated upon. What is of interest to cryptography is the first item. While it in many cases is possible to achieve the desired “traffic separation” by separate physical interfaces, or logically (implemented in routers), cryptographic separation is in many cases attractive as it is less expensive and more scalable than physical protection and offers true end-to-end security also against parties controlling the infrastructure. For this purpose, a number of security and key management protocols have been proposed in recent years, IPsec and IKE [5, 6, 3] perhaps being the most widely deployed ones.

While provable security was not main goals when designing these protocols, there has in the last few years been a trend to thoroughly study and analyze them, and it has been possible to prove security in some cases, e.g. [1].

Considering a network security solution where thousands, or hundreds of thousands of simultaneous connections need to be protected, there has been a need to use very efficient cryptographic algorithms and to provide hardware crypto support.

Another problem that has been studied is the management of the security in a network domain. A typical issue is the mass distribution of keys to thousands of nodes to maintain security. Related to this is also study of solutions for multi- or broadcast security where source origin authentication and revocation in large groups is an issue, see e.g. [4, 2].

A special type of network that has been discussed in the last few years are the so-called *ad-hoc networks*, where typically users without any pre-existing security association meet in an ad-hoc fashion and wants to create a logical, secure network. These networks are often run on “thin” clients such as PDAs or cell-phones.

References

- [1] R. Canetti and H. Krawczyk: “Security Analysis of IKE’s Signature Based Key-Exchange Protocol”, Proceedings of Crypto ’02, LNCS 2442, pp. 143–161.
- [2] D. Halevy and A. Shamir: “The LSD Broadcast Encryption Scheme”, Proceedings of Crypto ’02, LNCS 2442, pp. 47–60.
- [3] D. Harkins and D. Carrel: “The Internet Key Exchange (IKE)”, IETF RFC 2409.
- [4] A. Perrig, R. Canetti, J. D. Tygar, and D. Song: “The TESLA Broadcast Authentication Protocol”, RSA Cryptobytes, Summer 2002.
- [5] S. Kent and R. Atkinson: “IP Authentication Header”, IETF RFC 2402.
- [6] S. Kent and R. Atkinson: “IP Encapsulating Security Payload (ESP)”, IETF RFC 2406.

2.2 Financial services

2.2.1 Digital wallet, electronic cash and micro-payments

Electronic payment in general represents a vast and heterogeneous topic. It encompasses various non-anonymous protocols involving payment tokens with symmetric and asymmetric cryptographic capabilities such as electronic wallets, bank cards, etc., but also digital cash protocols -where anonymity is an essential part of the requirements. In the sequel, we will mainly restrict ourselves to digital cash, which represents one of the most challenging research areas for the application of cryptologic techniques to electronic payment.

Digital cash can be roughly described as a payment system involving customers, banks and possibly additional trusted authorities, in which the exchange of coins and notes encountered in ordinary (non-electronic) payments is replaced by the exchange of numbers (digital coins) representing certain amounts of money. Signed data representing a certain amount of money are typically delivered to a customer A by his bank prior to any transaction, subject to the withdrawal of the corresponding amount from A’s account. Digital coins are then transferred from A to recipients B during payment transactions, and passed later on from recipients B to their bank in order for the corresponding amount to be credited on B’s account. Digital cash is required to inherit some distinctive properties of cash, such as the untraceability of payments. However, due to the fact that unlike physical objects (e.g. coins and notes) digital information can be easily duplicated, digital cash systems must incorporate mechanisms for avoiding any double or multiple spending.

David Chaum initiated research on anonymous electronic payment (digital cash) and candidate cryptologic mechanisms to instantiate it, in the early eighties [CFN88]. But the first offline digital cash system (i.e. requiring no involvement of any bank or third party during the payment transaction) was proposed by Chaum, Fiat and Naor in 1988 [Cha82], and the first really efficient system was introduced less than ten years ago, by Brands [Bra93]. These seminal contributions were later on reflected and developed in those deliverables of the european project Esprit CAFE (Conditional Access for Europe) related to anonymous payment.

Blind signature (a mechanism enabling Alice to collect Bob's non-repudiable signature on a single message of her choice, without revealing Bob any information about this message) represents the core cryptographic primitive underlying all the previously mentioned schemes. It allows in particular banks to sign digital coins delivered to a customer without being capable later on to trace any individual coin they signed. The theoretical study of the security of this cryptographic primitive was only undertaken very recently. The first schemes providing provable security (in the random oracle model) against additional forgeries (i.e. against attacks where the execution of the protocol corresponding to k blind signatures would enable Alice to obtain $k+1$ valid signatures) were proposed by Pointcheval and Stern in 96 [PS96]. This represents a first step towards the construction of provably secure digital cash schemes, i.e. schemes for which counterfeit money is provably impossible to forge.

The above mentioned schemes provide users with perfect anonymity. This obviously does not only bring advantages (in terms of protection of personal data), but also strong disadvantages (it facilitates criminal misuses such as money laundering and blackmailing). Therefore, recent research on digital cash aims at introducing revocable anonymity. A payment schemes satisfies that property if a trusted authority can uncover the anonymity of any transaction, subject to a legal request. This model was questioned by numerous searchers : how to prevent revocable anonymity from also introducing potential misuses, e.g. the technical feasibility of uncovering the anonymity of some transactions without any prior legal mandate? Pfitzmann and Sadeghi introduced the concept of self-escrowed cash (against user blackmailing). In this alternative paradigm, users act as their own trusted authority, thus keeping the capability to track digital coins they might have been forced to spend (fund extortion). Kueger and Vogt introduced the concept of auditable tracing , i.e. the requirement that users be a posteriori capable to know whether the anonymity of their transactions was uncovered or not - and thus to turn against the trusted authority in case of undue revocations.

References

- [Bra93] S. Brands, Untraceable off-line cash in wallets with observers, Proceedings of CRYPTO'93, Lecture Notes in Computer Science, vol. 773, Springer-Verlag, pp. 302-318.
- [CFN88] D. Chaum, A. Fiat et M. Naor, Untraceable electronic cash, Proceedings of CRYPTO'88, Lecture Notes in Computer Science, vol. 403, Springer-Verlag, pp. 319-327.
- [Cha82] D. Chaum, Blind signatures for untraceable payments, Proceedings of CRYPTO'82, Plenum Publishing, pp. 199-203.
- [PS96] D. Pointcheval et J. Stern, Provably secure blind signature schemes, Proceedings of ASIACRYPT'96, Lecture Notes in Computer Science, vol. 1163, Springer-Verlag, pp. 252-265.

2.3 Social aspects

2.3.1 Ambient intelligence

Definition. Ambient Intelligence (AmI) is the vision that technology will become invisible, embedded in our natural surroundings, present whenever we need it, enabled by simple and effortless interactions, adaptive to users and context and autonomously acting. High quality information must be available to any user, anywhere, at any time, and on any device.

The embedding of computing and networking capabilities into more and more objects results in new and severe demands onto electronic devices in terms of functionality, design, power, robustness, wireless communication, packaging, and also cost.

Technically speaking this vision comes down to a possibly large network of small and compact devices, all interconnected with each other through wireless communication channels. Due to the highly dynamic nature of this networks, these communication channels will be setup in an ad-hoc fashion, possibly requiring multiple hops to setup a connection. Examples of these intelligent networks include environmental control in office buildings, monitoring of integrity of civil structures, robot control and guidance in automated manufacturing environments, warehouse inventory, integrated patient monitoring, diagnostics, and drug administration in hospitals, interactive toys, and the smart house providing identification, personalization and security.

Challenges. The security requirements for ambient intelligent networks are largely comparable to those of earlier networks. However, there are some important differences:

1. Because of the miniature size of the network nodes, the resources (CPU power, RAM, energy) available are very restricted. This means that protocols that make excessive use of energy consuming cryptographic primitives (e.g., RSA public key cryptography) cannot be used.
2. Denial-of-service attacks become increasingly important because the implications are not only loss of network connection and services, but possibly permanent shutdown of the node (due to battery exhaustion [10, 9]).
3. The dynamic and ad-hoc nature of the network makes key management one of the largest challenges in security design for ad-hoc networks.
4. Individual nodes are easily accessible. This means that theft of nodes is easy and implies that new security protocols should be designed so that they can cope with disclosure of secrets contained in these nodes.
5. Location privacy becomes an important issue for nodes that are carried around by users.

State-of-the-art.

Routing Security A number of routing protocols for ad hoc networks have been proposed, among which the Ad hoc On-demand Distance Vector (AODV) protocol and the Dynamic Source Routing (DSR) protocol. Unfortunately these protocols do not address

security issues. A number of “rescue” efforts have emerged as a result: (1) Marti *et al.* [6] pioneer the idea of the *wathdog* and *pathrater* that monitor the behaviour of other nodes in the network; (2) Along the same line of investigation are Buttyan *et al.* [2] who conceptualize the motivation for nodes not be selfish (regarding forwarding packets for other nodes) as *nuglets*, a sort of virtual currency. (3) SPINS (Security Protocols for Sensor Networks) provides broadcast security by using symmetric cryptography alone [7]. The target wireless network is static and homogenous, and the protocol relies on a trusted central base station.

Key Management Basagni *et al.* [1] reason that since sensor networks are so resource-constrained that only symmetric key cryptography is feasible, it is inevitable that clusters of nodes share a symmetric key, and on a network-wide level, all *pebblenets* share a traffic encryption key.

Zhou and Haas [12] propose a key management service that is distributed over $t + 1$ servers among n nodes, so that at most t nodes may be compromised, by the principle of threshold cryptography. A key management server not only has to store its own key pair, but also the public keys of all nodes in the network.

Hubaux *et al.* [5] go a step further by requiring each node to maintain its own certificate repository. These repositories store the public certificates the nodes themselves issue, and a selected set of certificates issued by others.

Efficiency A number of solutions have been proposed to cope with the energy and communicational restraints. Rich Uncle protocols have been proposed to off-load demanding tasks to more resourceful nodes [3]. Other proposals like [8] try to replace expensive asymmetric primitives with more efficient primitives based on symmetric cryptography.

References

- [1] S. Basagni, K. Herrin, D. Bruschi, and E. Rosti. Secure pebblenets. In *In Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 177–228, oct 2001.
- [2] L. Buttyan and J.-P. Hubaux. Nuglets: A virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. Technical Report DSC/2001/001, Department of Communication Systems, Swiss Federal Institute of Technology, 2001.
- [3] D.W. Carman, P.S. Kruus, and B.J. Matt. Constraints and approaches for distributed sensor network security (draft). Technical report #00-010, NAI Labs, June 2000.
- [4] Y.-C. Hu, A. Perrig, and D. B. Johnson. Wormhole detection in wireless ad hoc networks. Technical Report TR01-384, Department of Computer Science, Rice University, December 2001.
- [5] J.-P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001)*. ACM Press, 2001.

- [6] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in ad hoc networks. In *Proceedings of Mobile Computing and Networking (MOBICOM '00)*, pages 255–265, 2000.
- [7] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Seventh Annual International Conference on Mobile Computing and Networks (MobiCOM 2001)*, July 2001.
- [8] L. Reyzin and N. Reyzin. Better than ba: Short One-Time Signatures with Fast Signing and Verifying. In Jennifer Seberry, editor, *Proceedings of Information Security and Privacy – 7th Australasian Conference (ACSIP '02)*, Lecture Notes in Computer Science. Springer-Verlag, 2002.
- [9] F. Stajano. The resurrecting duckling – what next? In B. Christianson, B. Crispo, J.A. Malcolm, and M. Roe, editors, *Proceedings of the 8th International Workshop on Security Protocols*, Lecture Notes in Computer Science 2133, pages 204–214. Springer-Verlag, 2000.
- [10] F. Stajano and R. Anderson. The resurrecting duckling: Security issues in ad-hoc wireless networks. In B. Christianson, B. Crispo, and M. Roe, editors, *Proceedings of the 7th International Workshop on Security Protocols*, Lecture Notes in Computer Science. Springer-Verlag, 1999.
- [11] Y. Zhang, W. Lee, and Y. Huang. Intrusion detection techniques for mobile wireless networks. In *Mobile Networks and Applications (MONET)*. ACM/Kluwer, 2002.
- [12] L. Zhou and Z. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 13, no.6, November/December 1999.

2.3.2 Timestamping

Digital time-stamping is a set of techniques that enables us to determine if a certain digital document has been created or signed before a given time. In most practical applications, the time-stamping service is performed by a trusted third party – a Time-Stamping Authority (TSA) – that creates time-stamps. These time-stamps are the digital assertions that a given document was presented to the TSA at a given time. To get a document time-stamped, a client sends a request, containing a hash value of his document, such that the content of the document remains undisclosed for the TSA. A third party can verify the time-stamp on a given document, in some cases with the cooperation of the TSA.

Establishing the time of existence for digital documents is of high importance for some applications such as the signing of electronic contracts. It is also important for EDI (Electronic Data Interchange which is essential for electronic trade), IPR (Intellectual Property Rights) and interactive multimedia services (pay-TV, homeshopping, video online, ...).

Time-stamps are also essential as evidence in services providing non-repudiation of digital signatures. The basic problem in this case is that all certificates will expire or be revoked at some point in time. This implies that all signatures, generated with the corresponding private key will then become invalid. Besides the purely practical inconvenience, there also

exists a possible threat for non-repudiation: a signer might intentionally disclose his private signature key and then claim that previously made signatures were forged. Therefore one must be able to determine at a later time if a document was signed within the validity period (and before revocation) of the signature key, using the corresponding certificate or certificate revocation list to establish the validity period or revocation time. This way timestamps offer the possibility to generate signatures that are valid for a very long period. Practical issues, like algorithms that become less secure, can be solved by time-stamping again with stronger algorithms.

Several mechanisms to implement a time-stamping service exist, requiring different degrees of trust in the TSA offering the service.

Simple schemes produce time-stamps that are independent of one another, using either public key or symmetric key cryptography: the TSA adds the time information to the request and then computes a digital signature or message authentication code over the result. The verification procedure consists of checking the time parameter, the digital signature (with the TSA's public key) or message authentication code (for this the TSA is needed) and the correspondence between the document and the hash value in the request. The drawback of these methods is that the TSA must be trusted completely: it can produce false time-stamps without this being detected. Because a time-stamping service is supposed to offer long-term security it would be better to reduce the level of trust required in the TSA.

Linking schemes try to lower the required trust in the TSA by producing time-stamps which are linked to one another. Linking happens in three phases:

Aggregation: in the first step, all documents received by the TSA within a small time interval – the aggregation round – are being considered simultaneous. The output of the aggregation round is a binary string that securely depends on all the documents submitted in that round. Users receive information on how to compute the aggregation output, using their submitted document. The purpose of aggregation is to lower the load on the TSA, if the linking operation is expensive.

Linking: the output of the aggregation round is taken, and linked to previous aggregation round values, where the output of the linking operation cannot be computed without previous aggregation round values. This establishes a one-way order between aggregation round values, such that so-called *relative temporal authentication* is obtained: time-stamps of different aggregation rounds can be compared. This implies also that a time value is not necessary in linking schemes.

Publication: From time to time (e.g., each week), the TSA publishes the most recent time-stamp in a widely witnessed medium, such as a newspaper. By doing this, the TSA commits itself to all of the previously issued time-stamps. The published values are used for verifying time-stamps and they enable other parties to check if the TSA is behaving properly.

Linking schemes in the context of time-stamping were first described by Haber and Stornetta in 1991 [1, 2]. In 1998, alternative linking schemes were suggested by Buldas *et al.* [3],

and in 2000, they were optimized in time-stamp size [6].

Distributed Schemes. Another way of lowering the required level of trust in the TSA is to distribute the trust. In that approach, multiple users/TSAs cooperate to generate a time-stamp, possibly using a secure distribution of secret data necessary to generate a time-stamp. In this way, forgery of a time-stamp requires the collusion of a predetermined (high) number of parties, which is considered to be very unlikely. Examples of such protocols were first proposed by Benaloh and de Mare in 1991 [5, 4], and later on by Ansper *et al.* in 2001 [7]. They can be extensions of the schemes described above, and as such provide relative temporal authentication or absolute temporal authentication.

The theory behind digital time-stamping has only been developed since the 90's so it is still a fairly new topic in cryptography. Surety, Inc. is a pioneering company holding several patents in this area, and in the last couple of years several other companies have begun to offer time-stamping services. A large growth can be expected in this field for the near future, because the legal significance of digital documents is only starting to be recognized now (several countries have adopted new laws giving digital signatures the same legal value as their paper equivalents).

References

- [1] S. Haber, W. S. Stornetta, "How to Time-Stamp a Digital Document," *Journal of Cryptology*, Vol. 3, No. 2, 1991, pp. 99–111.
- [2] Dave Bayer, Stuart Haber, W. Scott Stornetta, "Improving the Efficiency and Reliability of Digital Time-Stamping," *Sequences II: Methods in Communication, Security and Computer Science*, R. Capocelli, A. De Santis, and U. Vaccaro, Eds., Springer-Verlag, 1993, pp. 329–334.
- [3] Ahto Buldas, Peeter Laud, Helger Lipmaa, Jan Willemson, "Time-Stamping with Binary Linking Schemes," *Advances on Cryptology - CRYPTO '98*, LNCS 1462, Hugo Krawczyk, Ed., Springer-Verlag, 1998, pp. 486–501.
- [4] J. Benaloh, M. de Mare, "One-way Accumulators: A Decentralized Alternative to Digital Signatures," *Advances in Cryptology - Proceedings of EuroCrypt '93*, LNCS 765, T. Helleseeth, Ed., Springer-Verlag, 1993, pp. 274–285.
- [5] J. Benaloh, M. de Mare, "Efficient Broadcast Time-Stamping," *Technical report TR-MCS-91-1*, Clarkson University, Department of Mathematics and Computer Science, April 1991.
- [6] Ahto Buldas, Helger Lipmaa, Berry Schoenmakers, "Optimally Efficient Accountable Time-Stamping," *Public Key Cryptography - PKC'2000*, LNCS 1751, Springer-Verlag, 2000, pp. 293–305.
- [7] Arne Ansper, Ahto Buldas, Märt Saarepera, Jan Willemson, "Improving the availability of time-stamping services," *The 6th Australasian Conference on Information Security and Privacy - ACISP'2001*, LNCS 2119, Springer-Verlag, 2001, pp. 360–375.

2.3.3 E-voting

Definition. An electronic voting scheme is a set of protocols which allows voters to cast ballots while a group of authorities collects the votes and outputs the final tally.

Voting schemes are thus defined by a registration protocol, during which voters register and may get/provide some information for authenticating themselves later; a voting protocol, during which a voter casts his vote; and a counting protocol which produces the final tally. The latter protocol may be run by either any party, or by authorities only. A verification protocol may also be available to allow anybody, or trusted parties only, to check the validity of the tally.

Until a few years ago, the “yes/no” paradigm in which voters can only cast a boolean vote has been used for technical reasons. But this model is not really practical since one usually wishes to further consider the null vote at least. Moreover, in a practical system, the tally should be computed at different levels in order to give local, regional and national results; and multiple candidates with the possibility of partial tally computation would be preferable.

A voting scheme is said to be universally verifiable if anybody can check any step of all protocols, until the final tally. Then, anybody can be guaranteed that all the votes have been considered, without any alteration. However, in such a voting scheme, anonymity of voters, or confidentiality of the votes, are usually achieved in a computational sense only, but not in the information theoretical sense. This means that, maybe, a few years later, one may know who votes to whom. If one stresses with long term confidentiality, one needs to consider information theoretical confidentiality, and then only individual verifiability is provided: anybody can check that his own vote had been taken into account.

State-of-the-art. Election schemes were first described by Chaum in 1982, and Benaloh in 1985. Most of the voting schemes proposed thereafter discussed only “yes/no” votes, until recently. Four different kinds of technologies have been proposed so far, according to the type of cryptographic primitive used:

Homomorphic encryption An homomorphic (public-key) encryption scheme is an encryption scheme (E, D) from one group structure $(G_1, +)$ into another group structure $(G_2, *)$ such that $D(E(m_1) * E(m_2)) = m_1 + m_2$. In the voting scheme, all voters send their encrypted votes to a single combiner. Such an encrypted vote is published on a bulletin-board: using the homomorphic property of the cryptosystem, the combiner but also anybody can compute the encrypted tally. Then, $(t + 1)$ out of the authorities can recover the tally by running a threshold cryptosystem. This model is optimal for the communications between voters and authorities. Such a technology, combined with zero-knowledge proofs, provides a universally verifiable voting scheme. Zero-knowledge proofs are indeed central tools to provide universal verifiability still keeping confidentiality of votes.

Blind signatures Blind signature schemes have been proposed by Chaum in 1981 to provide anonymity, essentially for electronic cash. A blind signature scheme allows a user to get a chosen message signed by a server in such a way that the server does not know the message and the signature either (no information about any of them).

With such a primitive, together with an anonymous channel, it is possible to describe an anonymous voting scheme: the voter interacts with the server to get his vote signed by the server (the server authorizes only one interaction with each voter). Then, he can send anonymously (via the anonymous channel) his vote with the server's signature which guarantees the validity of the vote. Such a signed vote is published on a bulletin-board: anybody can thereafter compute the tally.

The main problem with voting schemes based on blind signatures is that it is not universally verifiable, but individually only: anybody can check that his own vote is on the bulletin-board, but cannot be sure that others' votes have not been modified by the authority (who can sign any vote!). On the other hand, depending on the blind signature scheme, unconditional anonymity is possible. Blind signatures are interesting tools in cryptography, with many other applications, whenever anonymity is required.

Mix-networks A mix-net is a primitive that receives a list of ciphertexts and outputs all the corresponding plaintexts, but without leaking any information about which plaintext corresponds to which ciphertext. However, it further provides the guarantee that all the ciphertexts have been decrypted, using zero-knowledge proofs.

Such a primitive allows one to use any encryption scheme (without homomorphic properties): the voters encrypt and sign their votes; the mix-networks decrypt all the votes, and additionally provide a proof that the decryptions are all correct. Thereafter, anybody can compute the tally.

Such a technology also provides a universally verifiable voting scheme, but mix-networks are not efficient. Furthermore, many weaknesses have been discovered on schemes based on mix-nets.

Cryptographic counters Cryptographic counters are encrypted counters such that any voter can increase or decrease. Other participants have no information about the modifications introduced.

Eventually, an authority can decrypt the counters to get the final tally. Homomorphic encryption schemes are particular cases of cryptographic counters.

New trends. To achieve universal verifiability, the communication model in use is a public broadcast channel with memory, which can be implemented with a bulletin-board: all communication with the bulletin board are public and can be universally monitored. No party can erase any information but each voter can enter his part of the board.

Election schemes require the following properties:

- Eligibility: Only the authorized participants can vote.
- Privacy: The privacy of users ensures that a vote will be kept secret from any t-coalition of authorities.
- Universal verifiability: This ensures that any party including observers can convince himself that the election is fair and that the published tally has been correctly computed from the ballots that were correctly cast. A weaker notion is individual verifiability, where any voter can check that his own vote has been considered in the tally.

- **Robustness:** The robustness of the scheme ensures that the system can tolerate some faulty authorities who try to cheat during the computation of the tally.
- **Anonymity:** The votes cast by voters should be hidden.
- **Receipt-Freeness:** The receipt-freeness property ensures that a voter must not be able to construct a receipt proving the content of his vote. A weaker notion is incoercibility, where nobody can enforce someone to cast a specific vote (if he does not want, the voter can vote in such a way that no receipt is available).
- **Fairness:** No partial result will never be known before the final tally can be computed.

Any faulty voter or authority should be detectable.

Another kind of attack is possible in electronic systems, the so-called "Rushing Attack": At the closure time of the voting system, the local authorities reveal their local tally. The system can be protected to withstand a rushing attack of users who try to falsify the tally if they wait the result of the local authority to vote.

For a few years, several voting schemes have been proposed, with security analyses proving strong security properties, even receipt-freeness under some physical assumptions: the privacy, or anonymity, of the votes is guaranteed with proof of unforgeability, however, the final tally is universally verifiable. Furthermore, nobody can produce a receipt that would help him to convince someone of the contents of his vote (in case he would like to sell it).

Unfortunately, all the security proofs only provide security arguments, and not unconditional proofs. They further require additional physical assumptions which may not be realistic. First, they often rely on idealized assumptions, as required for proving the security of public-key encryption schemes, signature schemes and non-interactive zero-knowledge proofs of knowledge or membership, which are some building blocks of voting schemes. Second, private or anonymous channels are often required, with blind signature, or to achieve receipt-freeness. This does not seem that much realistic in practice.

Therefore, the main goals of future work is to manage to use primitives (encryption, signatures, ...) that can be proved in the standard model, without any idealized assumption. Furthermore, the efficiency of the reduction has to be made as good as possible, and the assumptions have to be realistic.

The past 15 years have seen the emergence of various security notions (anonymity, incoercibility, receipt-freeness, etc.). In the long term, one would like to know if such notions are the right ones, or if one must achieve stronger security notions, and if methods to achieve such security notions will evolve.

Indeed, four main methods have been proposed so far, but none is fully acceptable (both from the security and efficiency points of view). Therefore, one also needs alternative methods, and maybe new computational assumptions.

References

- [1] M. Hirt and K. Sako. Efficient Receipt-Free Voting Based on Homomorphic Encryption. Eurocrypt'00, pp. 539-556, LNCS 1807, Springer, 2000.

- [2] A. Fujioka, T. Okamoto and K. Ohta. A Practical Secret Voting Scheme for Large Scale Election. Auscrypt'92, pp. 248-259, LNCS 718, Springer, 1993.
- [3] Y. Desmedt and K. Kurosawa. How to Break a Practical MIX and Design a New One. Eurocrypt'00, pp. 557-572, LNCS 1807, Springer, 2000.
- [4] J. Katz, S. Myers, and R. Ostrovsky. Cryptographic Counters and Applications to Electronic Voting. Eurocrypt'01, pp. 78-92, LNCS 2045, Springer, 2001.

2.3.4 Digital Rights Management

Definition. In this section a definition of Digital Rights Management (DRM) is given and its goals are described. For more detailed information about DRM, the reader is referred to [1].

Whenever content (*i.e.* information) has been created, the owner of this content has specific rights associated with it. These rights are usually divided into three types:

Legal: the rights the owners have under law, *e.g.* copyright or rights obtained by patents.

Transactional: the rights that are obtained or lost by selling or buying them, *e.g.* by buying CDs or when an artist sells his/her music to a record label/company.

Implicit: the rights that are derived from the information carrier and the format of the information, *e.g.* copying a book without degrading its quality and distributing these copies are expensive and time-consuming tasks.

Nowadays, more and more content becomes available in digital formats. With respect to the management of rights associated to this content, the two main differences with traditional formats like *e.g.* books are the following. Firstly, copies can be made without any degradation in quality, *i.e.* these copies are as good as the original. Secondly, the widespread use of the Internet enables a fast and cheap distribution of digital content; the use of digital information carriers that are slower and more expensive to distribute, like *e.g.* CDs, DVDs or Floppy Discs, can be avoided. Note that both these aspects are caused by changes of the implicit rights of the information (see Type 3 in the list mentioned above). These changes imply that adversaries can modify, copy and distribute digital information in ways that were not possible or too expensive for traditional media formats. Consequently, providers of digital content (like the movie and music industry) need technologies to protect the rights associated with this content. Digital Rights Management (DRM) is a technology for controlling and managing these rights.

The main goal of DRM is to make digital content only accessible to users who own the rights for this. If content owners want to exploit the benefits of the Internet (cheap and fast distribution, new business models), DRM is needed to avoid piracy by enforcing that the digital content can only be accessed as specified by the owner. It protects the rights of the content provider/owner, and avoids a potential loss of income caused by illegal copying, modification and re-distribution by adversaries.

State-of-the-art and new trends.

Tamper-resistant software/hardware. The security model for DRM systems differs from more 'conventional' security models used in other cryptographic systems, in the sense that the end-users are not trusted in the DRM security model. In this model, the security of a system strongly relies on its robust implementation at the client side. A well-known technology that can be applied for achieving this is the use of tamper-resistant software at the client side (e.g. based on so-called code or program obfuscation). At this moment this technology is widely used, as many DRM solutions are based on existing hardware like PCs and adding / integrating new (tamper-resistant) hardware is often considered as being too expensive. However, recent theoretical work on the problem of program obfuscation [2] shows that this is impossible under a certain notion of obfuscation. Therefore it is expected that more future DRM solutions will be based on the use of tamper-resistant hardware at the client side to store secret information (cryptographic keys) that should not be accessible to the end-user and to perform certain computations with this secret information.

Cryptographic primitives and protocols. The following cryptographic building blocks are typically used in DRM systems. Identification of the user or his/her device to the server is needed in order to give this user or device the specified rights associated with the digital content. Examples of techniques used for this are cookies, digital certificates and serial numbers of processors. Other well-known solutions are based on a User ID - password combination, challenge-response identification and zero-knowledge protocols. Public-key cryptosystems (PKCs) like e.g. RSA are a common tool used for digital certificates. PKCs are also used in DRM systems for digital signatures and the encryption of short (i.e. symmetric) keys. DRM systems usually use symmetric encryption algorithms (e.g. DES/AES) for copy protection, i.e. the decrypted content is only accessible to the users who have the corresponding rights (or equivalently, own the corresponding symmetric key). In addition to using the symmetric algorithms for access control, they are also used for computing so-called Message Authentication Codes (MACs), which protect the authenticity of the content. These cryptographic primitives and building blocks are used in many applications including DRM, and are not specifically developed for DRM systems.

Watermarking and fingerprinting. Watermarking and fingerprinting play an important role in DRM, as the digital content will eventually be available to the (legitimate) user in decrypted format, opening possibilities for adversaries to illegally copy, modify and re-distribute such unencrypted content. Watermarks are used to embed information (like e.g. information about the content owner) in the digital content. Watermarks should be hard to remove or robust (e.g. compression or format conversion should not remove the watermark), easy-to-detect, imperceptible and reliable. Fingerprints are user-specific watermarks that are contained in the digital content and can be used to trace back malicious users. Fingerprints should have the additional properties that they are collusion resistant, frameproof and that malicious users are traceable. The collusion resistance property made fingerprinting a topic in cryptology. An existing theory that can be used for fingerprinting is based on codes with the so-called Identifiable Parent Property (IPP). Several construction methods for these codes with IPP and theoretical bounds on their parameters exist. However, so far the constructions are not too practical

in the sense that the size of a fingerprint has to be long in order to obtain a sufficient level of collusion resistance. A trend is 'asymmetric fingerprinting', where the owner can find information about the traitor comparable to a signature from the traitor of the fact that he/she misbehaved. Another trend is the addition of anonymity [3], where buyers can buy fingerprinted digital content without revealing their identity. However, as soon as a buyer misbehaves by redistributing the content, his/her identity can be revealed. Anonymity can play an important role in the public acceptance of DRM systems, as buyers might not want to give up their privacy (only) for the purpose of generating a fingerprint.

Key management issues. An important aspect of DRM systems is the key management schemes they use. Many DRM applications can use existing techniques not especially developed for that purpose. In addition, key management techniques that are more specific for DRM have been developed, such as those used in broadcast encryption (BE) schemes. Such schemes were introduced in the early nineties. An important issue in BE schemes is key revocation, as ideally it should be possible to remove compromised equipment from the system in an easy way and without affecting the security of the other components of the system. In addition, key revocation is needed whenever a user leaves the system. In BE systems it is usually desirable to be able to send encrypted messages to an arbitrary set of recipients while remaining secure in the event that a (predefined) number of users collude and share their secret information. Schemes that can deal with coalitions of any size are called 'long-lived' BE schemes [4]. The purpose of such schemes is to continue the broadcast in a secure way to authorised users in the case that some keys in the system have been compromised. Combinations of broadcast schemes with traitor tracing schemes have also attracted some attention.

References

- [1] B. Rosenblatt, B. Trippe and S. Mooney. *Digital Rights Management: Business and Technology*, John Wiley & Sons, 2001.
- [2] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan and K. Yang. On the (Im)possibility of Obfuscating Programs, *Advances in Cryptology - CRYPTO 2001* (J. Kilian, ed.), LNCS 2139, Springer (2001), pp. 1-18.
- [3] B. Pfitzmann and M. Waidner. Anonymous fingerprinting, *Advances in Cryptology - EUROCRYPT '97* (w. Fumy, ed.), LNCS 1233, Springer (1997), pp.88-102.
- [4] J. A. Garay, J. Staddon and A. Wool. Long-Lived Broadcast Encryption, *Advances in Cryptology - CRYPTO 2000* (M. Bellare, ed.), LNCS 1880, Springer (2000), pp. 333-352.

3 Cryptographic protocols

3.1 Two party protocols

3.1.1 On-line authentication (identification)

Overview. Entity authentication is a process where a *verifier* can be convinced of the identity of a *prover*. Entity authentication in the Information Society proceeds from the hypothesis that both the prover holds some specific digital secret data and that prover and verifier only communicate digital information. Three components are needed.

- The real-world identity of the prover needs to be able to generate some secret digital data that the prover can use.
- The secret digital data of the prover needs to be linked to some digital data that the verifier knows to correspond to the specific prover.
- These digital data are used to make an (interactive) proof that can convince the verifier that the prover holds the secret.

The first component is sometimes realized with personal devices (smart cards or badges), sometimes with biometrics, sometimes with a password. Depending on the context, it might be important that the identity cannot be copied (which excludes password) or cannot be transferred (which excludes personal devices). The cryptographic requirement for this component is that the entropy of the resulting digital secret is high.

The second component depends on key management and public key infrastructures (cf. the relevant sections of this document).

The third component is the main cryptographic component and is usually named *identification protocol*. Currently two approaches can bring some security. Challenge-responses protocols are MAC-based or digital signature-based and are widely used in practice. Zero-knowledge protocols have provable security but may have some practical disadvantages [3].

Secret generation. There is a renewed interest in entity authentication and key establishment schemes based on passwords: these are schemes where the prover can memorize a secret that has limited but sufficient entropy and that is easy to remember. Other trends are the extension of schemes based on passphrases [5], visual passwords, . . .

Biometrics are intensively studied, but most of this topic is not a cryptographic issue. However, there has been some recent work [4, 6] on deriving a secret from a biometric (such as a fingerprint) combined with an error-correcting code.

Identification protocols. The security model for challenge-response protocols and the way they are built from some components (MAC or digital signatures) is an active research area.

The current trend for identification protocols is to consider new attack models (e.g., concurrent attacks, reset attacks [1]) where some security proofs can be provided. It turns out

that secure identification does not need zero-knowledge, and that some protocols which are not zero-knowledge are more robust w.r.t. new attack models [2].

Off-line/on-line variants and server-aided variants are studied to improve the practical performance of zero-knowledge protocols and increase their use.

References

- [1] Mihir Bellare, Marc Fischlin, Shafi Goldwasser, Silvio Micali. Identification Protocols Secure against Reset Attacks. *Advances in Cryptology - Eurocrypt 2001 Proceedings, Lecture Notes in Computer Science Vol. 2045*, B. Pfitzmann ed, Springer-Verlag, 2001, pp. 495–511.
- [2] Mihir Bellare, Adriana Palacio. GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks. *Advances in Cryptology - Crypto 2002 Proceedings, Lecture Notes in Computer Science Vol. 2442*, M. Yung ed, Springer-Verlag, 2002, pp. 162-177.
- [3] Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems, *Advances in cryptology, Proc. CRYPTO '86, January 1987*, pp. 186–194.
- [4] Ari Juels and Martin Wattenberg. A Fuzzy Commitment Scheme, *ACM Conference on Computer and Communications Security, 1999*, pp. 28–36.
- [5] <http://cm.bell-labs.com/who/philmac/pak.html>.
- [6] C. Soutar and G.J. Tomko. Secure private key generation using a fingerprint. In *CardTech/SecurTech Conference Proceedings, Vol. 1*, pages 245–252, May 1996.

3.1.2 Off-line authentication (of a document)

Like for on-line authentication, there are two main techniques used to authenticate a document. The technique from symmetric cryptography is message authentication code (MACs) (see Section 4.1.6) where the two parties share a common secret. The technique from asymmetric cryptography is digital signature (see Section 4.2.2) where the signer uses his private key and the verifier uses the corresponding public key.

3.1.3 Secure communication channel

One of the oldest problem in cryptography is the establishment of a secure communication channel. The establishment of a secure communication channel is usually split in two phases. First, the two parties agree on an encryption/decryption technique. Then, one party sends encrypted data that the other one only can decrypt.

The two parties usually agree publicly on a fast symmetric encryption/decryption technique and then agree on a secret key. This secret key can be transmitted by another secure

channel, or by an asymmetric encryption scheme (see Section 4.2.1). Ad hoc key agreement techniques also exist, like the famous Diffie-Hellman scheme, and an ongoing research direction is the generalisation of two-party key agreement protocols to multiparty protocols.

For efficient encryption of data, two categories of cryptographic primitives are used: stream ciphers (see Section 4.1.2), which generate a random-looking bitstream that blinds the message with a XOR; and block ciphers (see Section 4.1.1) which are keyed substitutions of data blocks.

To protect against the man-in-the-middle attack, all protocols that setup a secure communication channel should have some authentication. There is ongoing research on techniques that achieve authenticated encryption faster than authentication plus encryption.

3.2 Multiparty protocols

3.2.1 Multiparty computation

In multiparty computation several users compute together some information. This could be a common session key or a joint secret of which the parties hold shares. One example hereof are threshold schemes where a certain number of shares is sufficient to compute the joint secret or unsecure networks that are used to compute involving private information. In the past years several scenarios have been studied solving *e.g.* linear algebra problems or to compute approximations.

However, none of the currently available multiparty computation methods is efficient. The existing techniques can serve as a proof-of-concept for the approach but are prohibitively expensive to be of any practical use. For example, since the approach is generic, all computations have to be formulated in terms of an unlocked digital circuit which is then jointly evaluated. And the amount of interaction among the parties is typically directly proportional to the depth of the circuit. Another limiting feature is that today's techniques rely on synchronized networks that provide the abstraction of a broadcast channel, which is not suitable for wide-area networks such as the Internet.

Recent work has shown how to make multiparty computation more practical in two directions: (1) minimizing the amount of interaction and (2) reducing the assumptions on the network.

1. So-called “optimistic protocols” run very fast, and with a minimal amount of interaction, when no faults occur. Recent examples of this approach include the multiparty computation protocol of Hirt, Maurer, Przydatek (Asiacrypt 2000) and the optimistic fair twoparty computation by Cachin and Camenisch (Crypto 2000).
2. The assumption of a synchronous network can be justified by clock synchronization protocols and that of a broadcast channel by a suitable Byzantine agreement protocol. However, a synchronous model involves making time-out assumptions, which is a delicate issue, because if wrong assumptions are made, the performance of the system is severely degraded. For example, if aggressively small time-outs are used and the network happens to slow down for some reason, then many parties will no longer be able to reach each other and declare that their peers to have exhibited a failure. One solution

to this problem is to avoid timing assumptions, which can be invalidated, and to use an asynchronous system model. Recent work has shown how practical agreement and broadcast protocols can be constructed in the asynchronous model (Cachin, Kursawe, Petzold, Shoup; Crypto 2001); they open a way to efficient multiparty computation protocols that are suitable for wide-area networks.

References

- [1] M. Hirt, U. Maurer and B. Przydatek. Efficient Secure Multi-Party Computation. In *Proc. of Asiacrypt '00*, Lecture Notes in Computer Science volume 1976, pp 143–161, Springer Verlag, 2000.
- [2] C. Cachin and J. Camenisch. Optimistic Fair Secure Computation. In *Proc. of Crypto '00*, Lecture Notes in Computer Science volume 1880, pp 94–112, Springer Verlag, 2000.
- [3] C. Cachin, K. Kursawe, F. Petzold and V. Shoup. Secure and Efficient Asynchronous Broadcast Protocols. In *Proc. of Crypto '01*, Lecture Notes in Computer Science volume 2139, pp 524–541, Springer Verlag, 2001.

3.2.2 Formal Models of Security and Composability of Protocols

Composability of cryptographic protocols has been studied in the context of *self composability*, i. e., many executions of the same protocol run concurrently. Already in this setting protocols which can be proven secure for a single execution can become insecure, e. g. Zero knowledge proofs [2] or Byzantine agreement [3].

As a new trend, models of security were proposed which guarantee a *universal composability* [1, 4]. The new models build on the established definition of security by *simulateability*, where a real protocol with a real adversary is compared to an ideal functionality where a simulator with very limited capabilities has to mimic the effect of a real attack. Whenever the real protocol and the ideal functionality are indistinguishable one says that the protocol in question realizes the ideal functionality *securely*.

Beyond other approaches the new models introduce an additional interactive machine, the *environment machine*. This machine plays the role of a distinguisher between a real protocol and an ideal functionality. The environment machine may choose all inputs, read all outputs, advise the adversary, and give additional information to participants of the protocol at any time.

For protocols securely realizing an ideal functionality in this sense, a *composition theorem* holds. In a secure application which employs an ideal functionality, this functionality may be replaced by a secure realization without violating the overall security of the application. Important cryptographic primitives like public key encryption, key exchange or signatures can be realized in a universally composable manner.

References

- [1] Ran Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *42nd IEEE Symposium on Foundations of Computer Science: proceedings: October 14–17, 2001, Las Vegas, Nevada, USA*, pages 136–145. IEEE, pub-IEEE, 2001.
- [2] Oded Goldreich and Hugo Krawczyk. On the composition of Zero-Knowledge Proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.
- [3] Yehuda Lindell, Anna Lysyanskaya, and Tal Rabin. On the composition of authenticated byzantine agreement. In *Proceedings of 34th Annual ACM Symposium on Theory of Computing*, pages 514–523, Montreal, Canada, 2002. ACM.
- [4] Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proc. of IEEE Symposium on Security and Privacy*, pages 184–200, 2001.

3.2.3 Key management and key establishment

There is an ISO standard [2] on key management, which contains an extensive list of protocols. There is also an ongoing NIST project. This could be taken as a sign that this is a mature area with ongoing research but with some fixed basis. However, for a variety of reasons there exists a significant gap between the mechanisms described in this standard, the detailed mechanisms used in practice, and the mechanisms studied in the cryptographic protocol literature. There is a very long tradition of research on key establishment protocols; the Handbook of Applied Cryptography contains an overview and one of the first taxonomies of the area [4].

Extensive work has been performed on the application of formal methods and logics to cryptographic protocols (cf. special issue of Journal of Cryptology on this topic [3]). While this work is very interesting, and has brought to light the many complex facets of protocol design, one criticism one may have is that the properties of cryptographic primitives assumed in this approach (e.g., ‘encryption’ is always authenticated encryption) has been quite different from properties achieved by ‘real’ cryptographic primitives. Moreover, these techniques cannot deal with the more involved algebraic properties of many public-key establishment protocols.

Among the best known work in the cryptographic community in this area is that of Bellare and Rogaway [1] who present a complexity-theoretic communications model and formal definitions for secure symmetric-key 2 and 3-party mutual authentication and authenticated key establishment. In this model they offer provably secure solutions based on the existence of pseudo-random functions or pseudo-random permutations.

Key escrow and key recovery has been a very active area of research in the mid 1990ies. Besides the mathematical or technical problems to find secure protocols allowing these properties, the question whether a country should or should not impose such restrictions on the privacy of its inhabitants is a political and sociological one.

References

- [1] M. Bellare and P. Rogaway. Entity Authentication and key distribution. *Advances in Cryptology - Crypto 93 Proceedings, Lecture Notes in Computer Science Vol. 773*, D. Stinson ed, Springer-Verlag, 1994, pp. 232–249.
- [2] ISO/IEC 11770, Information technology – Security techniques – Key management – Part 1: Framework, 1996, Part 2: Mechanisms using asymmetric techniques, 1996, Part 3: Mechanisms using asymmetric techniques, 1999.
- [3] R. Kemmerer, C. Meadows and J. Millen. Three System for Cryptographic Protocol Analysis. *Journal of Cryptology*, Vol. 7, No. 2, pp. 79–130.
- [4] A. Menezes, P. van Oorschot, S.A. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1997.

3.2.4 Public Key Infrastructures

While open Public Key Infrastructures (PKIs) [1, 4] with certificates have not become as ubiquitous and widespread as envisaged, they still form the cornerstone for deployment of security services. It is clear that there will not be a single super-PKI, but a large network of PKI-islands with complex trust relationships; in spite of the alternatives proposed, it seems likely that the organizations that have been responsible for issuing identities in the physical world will probably gradually take on the same role in the on-line world.

The main research topics in the PKI world are

- trust models: hierarchical model, browser model, hub-and-spoke model, enterprise model, web of trust (PGP);
- path discovery: how to (automatically) establish a trust relationship in a complex graph;
- alternative models such as SPKI and SDSI [7] which are slowly finding their way to applications; and
- efficient and scalable revocation [5].

In the real world trust centers appeared but only a few survived. Moreover, it seems that people are still not aware of the impact of cryptology on daily life or accept keys without or with outdated certificates. The end users should be sensitized to check the validity of certificates of pages for home banking etc. Even though there are standards dealing with certification and automatic validation, in practice this still poses a problem as trust centers still differ too much. We still seem to be far away from certificates for keys of private users as trust centers are expensive and at the moment no-one is willing to spend money on encryption of say emails.

Many alternative solutions have been proposed to distribute public keys, such as identity based (ID-based) public key systems [6], self-certified public keys and implicitly certified-keys.

There has been recently a surge of interest in ID-based cryptology, as Boneh and Franklin [2] proposed a practical scheme based on non-degenerate bilinear maps between groups. In the last year, many papers on improvements and generalizations have appeared. This solves the problem that a user might wish to send ciphertexts to recipients that do not yet possess a private key. Furthermore, as the user chooses the public key of the recipient no certificates for this key are needed. However, the protocol relies heavily on a trusted third party, which has to provide the legitimate recipient with his private key; this is inherent to ID-based cryptology. Certainly this allows key escrow which may or may not be desired. Moreover, in practice one will have multiple trusted parties, each with their own domain parameters; the problem of securely identifying someone's public key has now been moved to the problem of securely identifying the domain to which this person belongs (e.g., yahoo, msn, wanadoo, . . .) and to identify the correct public parameters of this domain.

References

- [1] Carlisle Adams and Steve Lloyd. *Understanding Public-Key Infrastructure: Concepts, Standards and Deployment Considerations*. New Riders, 1999.
- [2] Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. *Advances in Cryptology, Crypto 2001*, Lecture Notes in Computer Science 2139, Springer-Verlag, 2001, pp. 213–229.
- [3] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest. Certificate Chain Discovery in SPKI/SDSI. *Journal of Computer Security*, in print (2002).
- [4] A. Menezes, P. van Oorschot, S.A. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1997.
- [5] Moni Naor and Kobbi Nissim, Certificate Revocation and Certificate Update, 7th USENIX Security Symposium, 1998, pp. 217–228.
- [6] A. Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology, Crypto'84*, LNCS 196, Springer-Verlag, 1985, pp. 47–53.
- [7] SDSI A Simple Distributed Security Infrastructure,
<http://theory.lcs.mit.edu/~cis/sdsi.html>

3.2.5 Privacy and anonymity

With computers becoming increasingly networked and transactions being more and more carried out electronically, the amount of data stored and processed becomes a threat to the privacy of individuals. While storing and processing data is unavoidable to conduct transactions, the amount can be reduced and thereby the threat lessened. Cryptography provides many technical means for this while at the same time increases the security. This thread has recently become widely acknowledged by now, and an area of privacy enhancing cryptographic protocols has obtained increasing attention in the recent years.

As a result many of the known primitives have been revisited and more efficient instantiation of them proposed and a number of new primitives. Also, while for other areas it was standard for quite some time to provide formal models and then prove the actually scheme secure w.r.t. to them, this caught on only recently in this area. Consequently, many of the known schemes were found to be flawed and better schemes were proposed. Examples are confirmer signatures scheme and anonymous payment systems. While for some of these primitives the right security definitions have been found, for others still more definitional work is required.

An important privacy-enhancing primitive are blind signature schemes. They are a protocol that allow users to obtain a signature from the signer without the signer learning the message that is signed. Even if the signer sees the message and the signature later on, she cannot link them to the protocol instance in which they were produced. Blind signature are used for instance to build anonymous payment and voting systems. In this area a number of new provably secure schemes were proposed that improve on the message complexity as well as on the number of signatures that can be issued. On the downside, all the practical schemes are proven secure in the random oracle model, which has attracted some critique on its validity.

Another ingredient that allows one to minimizing the data users need to provide to conduct secure transaction are anonymous credential systems. Using such a system, a user can provide certified attributes to her communication partners. An application of this is for instance an anonymous subscription to a service there the user can prove using an anonymous credential that she is eligible to the service without revealing her identity. In this area the first practical scheme were proposed only recently and further advances can be expected in the future.

Related to anonymous credential systems are group signatures and identity escrow schemes. They allow users to prove that they belong to some group that is managed by an authority. However, here anonymity is conditional, i.e., the central authority is able to revoke the reveal the identity of the user if necessary. An example of where such scheme are useful is anonymous access control: the user can prove that she is part of a group who is allowed to access are resource or to enter a building. When later it turns out that this access was related to some illegal activity, then the authority can nevertheless identify the culprit. A recent cousin of group signature are ring signatures. Here, a user can prove that she a members of set users that she specifies. In ring signatures the potential members of the group are known, where as this need not be the case for group signatures.

However, any of these cryptographic primitives does not achieve its goal if the underlying communication network is not anonymous. This is for instance the case for the Internet. A primitive that comes to the rescue are mix-net works. They consists of a number of servers that reroute (mix) network traffic and thereby anonymize it. While the concept of mix networks is known for a while, many proposed protocols have been found flawed. Mix networks belong also to the primitives where proposals are often only analyzed heuristically and no realistic formal model exists. Recent trends in this area are ad-hoc and peer-to-peer networks and measure for the amount of anonymity a certain (configuration of a) mix-network provides.

Finally, a recent trend in privacy enhancing technology is law enforcement. For some areas such as anonymous electronic cash is has since long been recognized that anonymity can be misused (von Solms and Naccache showed that anonymous e-cash is ideal to collect a ransom)

and counter measures have been developed. For other areas, technology that support privacy and law enforcement at the same time is still missing.

References

- [1] M. Abe. A Secure Three-Move Blind Signature Scheme for Polynomially Many Signatures. In *Proc. of Eurocrypt '01*, Lecture Notes in Computer Science volume 2045, Springer Verlag, 2001.
- [2] J. Camenisch and A. Lysyanskaya. Efficient Non-transferable Anonymous Multi-show Credential System with Optional Anonymity Revocation In *Proc. of Eurocrypt '01*, Lecture Notes in Computer Science volume 2045, Springer Verlag, 2001.
- [3] S. von Solms and D. Naccache. On Blind Signatures and Perfect Crimes. In *Computer & Security*, number 6, volume 11, pp 581–583, 1992.

3.2.6 Quantum Cryptographic Protocols

Quantum key exchange has been proven to be realizable in principle by several different physical realizations. The security of quantum key exchange is still under investigation. General proofs of security were given for models with idealized apparatus [5, 6], but concrete realizations allow more attacks [2].

The possibility of an unconditionally secure key exchange has led to a search for quantum protocols realizing other cryptographic primitives like bit commitment or oblivious transfer. The results have mostly been negative, especially bit commitment and oblivious transfer cannot be realized with unconditional security [4, 3]. But still quantum cryptography can achieve tasks classical cryptography cannot. Deniable key exchange [1] is an example or the possibility to obtain oblivious transfer from bit commitment [7], which allows to realize cryptographic protocols relative to weaker assumptions than possible classically.

References

- [1] Donald Beaver. On Deniability in Quantum Key Exchange. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages, 352–367, Amsterdam, The Netherlands, 2002. Springer.
- [2] Gilles Brassard, Norbert Lütkenhaus, Tal Mor, and Barry C. Sanders. Security Aspects of Practical Quantum Cryptography. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 289–299, Berlin, 2000. Springer.
- [3] Hoi-Kwong Lo and H. F. Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78:3410–3413, 1997.
- [4] Dominic Mayers. Unconditionally secure bit commitment is impossible. *Phys. Rev. Letters*, 78:3414–3417, 1997.

- [5] Dominic Mayers. Unconditional security in quantum cryptography. Available on the Los Alamos preprint archive at <http://xxx.lanl.gov> as quant-ph/9802025, February 1998.
- [6] Peter W. Shor and John Preskill. Simple Proof of Security of the BB84 Quantum Key Distribution Protocol. *Phys. Rev. Lett.*, 85:441–444, 2000.
- [7] Andrew Yao. Security of quantum protocols against coherent measurements. In *Proceedings of the 27th Symposium on the Theory of Computing*, pages 67–75. ACM, Las Vegas, June 1995.

4 Cryptographic techniques

4.1 Symmetric cryptography

4.1.1 Pseudo-random permutations: block ciphers

Blockciphers represent together with streamciphers one of the two main classes of primitives encountered in symmetric cryptology. A blockcipher can be described as a keyed pseudo-random permutation of the $\{0, 1\}^n$ set of n – bit blocks, whereas a streamcipher can be described as a keyed pseudo-random sequence over a finite alphabet (e.g. $\{0, 1\}$). The most usual blocklengths n for existing blockciphers are 64 to 128 bits.

Blockciphers are typically slower than stream ciphers (20-40 cycles/byte) and require more gates (5000-100,000). However, they form a very flexible building block, that can be used in various modes of operation for confidentiality, message or entity authentication, one-way functions, hash functions, etc. Blockciphers can even be efficiently converted to a streamcipher, if used in an appropriate mode of operation (such as OFB), whereas the converse is not true. Historically, blockciphers have been more prominent than streamciphers in open standards (DES, Triple-DES, AES), which may explain their popularity. Blockciphers are preferred wherever a flexible and standardized building block is needed. Therefore they are used in many cryptographic applications such as home banking, e-mail, authentication, key distribution and sometimes encryption in mobile telephony, in hard disk encryption, and so on, whereas stream ciphers are preferred for selected applications with high performance or low power requirements.

In the mid 1970ies, the blockcipher standard DES (Data Encryption Standard) was designed in a secret way and published by the US NBS (National Bureau of Standards, now NIST, National Institute for Standards and Technology), without explaining the underlying design principles. DES has been the de facto world standard for encryption until the mid 1990ies: in the last years the short key length of DES (56 bits) had undermined its security more and more. In critical applications DES was often replaced by Triple-DES (threefold iteration of DES), which is however three times slower. In addition, certain applications required a block length larger than 64 bits (both DES and Triple-DES operate on 64-bit blocks). In 1997, NIST refused to extend DES for another 5-year period, and the need for a new blockcipher standard was apparent. NIST announced a competition for selecting this standard in an open and transparent procedure. Out of the 15 algorithms submitted to this

competition by various academic and industrial laboratories, the winner, namely the Belgian proposal **Rijndael** by Rijmen and Daemen, was selected in 2000. Rijndael is now adopted as the AES (Advanced Encryption Standard). More than half of existing security products currently use DES or variants of DES. Many products will shift to AES and a large part of the confidentiality of mass market applications of the cryptology will be based on the security of AES. Outside from DES, Triple-DES and AES, several other recently proposed blockciphers are also used in numerous security products, for instance IDEA (an algorithm used in the PGP file encryption software), **RC5** (an algorithm used in many S/MIME protected email products), **MISTY1** and its variant **KASUMI** (which was adopted encryption and message authentication algorithm for the UMTS third generation mobile system), and numerous blockcipher proposals are currently being evaluated as part of the European project NESSIE.

Studies made during the 25 years of existence of DES have led to important theoretical advances in the public knowledge on the design of blockciphers. The discovery of **differential and linear cryptanalysis** techniques in the early 90's represent (together with precomputation techniques such as Hellman's Time-memory trade-off) the most significant advances in the analysis of DES and more generally of iterated blockciphers, so that the resistance to these attacks has become one of the main criteria in the analysis of the strength of blockciphers. Some recently proposed designs, e.g. MISTY and KASUMI (which nested structure exploits upper bounds of differential and linear transition probabilities established by Nyberg, Knudsen, Aoki, Matsui et al.), or constructions based upon the so-called decorrelation theory by Vaudenay, offer provable resistance against basic forms of differential and linear cryptanalysis. However, such proofs only show that when the average is taken over all keys, attacks are not feasible. When a blockcipher is being cryptanalysed, the opponent is trying to recover a particular key, and these proofs provide no guaranteed upper bound on the actual maximum transition probability obtained for each individual key. The blockcipher AES does not have such a proof of resistance against differential and linear cryptanalysis. However, its internal design is based on a certain strategy (the wide trail strategy) which provides some plausible argument of security against such attacks.

Several cryptanalytic methods other (and often more specific) than differential and linear cryptanalysis have been discovered over the past decade, for instance : higher order differential attacks, truncated differential attacks, interpolation attacks, integral (saturation) attacks, impossible differential, boomerang, and rectangle attacks which may in certain cases to distinguish up to two times more rounds from a random permutation than usual differential techniques, chi-square, partitioning, and stochastic cryptanalysis, and also attacks against key schedules, e.g. sliding attacks, or related key attacks. Although formal proofs of security against these various classes attacks have not been systematically developed for existing blockciphers, their existence is generally taken into account by the designers of blockcipher proposals, and an algorithm such as AES can be reasonably conjectured to resist these attacks techniques (most of which are essentially statistical in nature).

On the other hand, the only assertion one has for now is that for the time being, there exists no feasible shortcut attack on AES. Since AES is making use of several algebraic structures, it cannot be entirely precluded that further use of advanced algebraic techniques (Gröbner basis, probabilistic interpolation, quadratic approximations, genetic algorithms) might establish weaknesses in AES.

Outside from the study of various categories of attacks mentioned above and of design

methods to resist these attacks, the cryptologic research on blockciphers has been also strongly influenced over the past years by the development of unconditional security proof techniques allowing to partially validate one specific level of a blockcipher construction or alternatively a mode of operation of a blockcipher, namely the security paradigm proposed by Luby and Rackoff in 1988. In this security paradigm (which was later on developed by Patarin, Maurer, Rogaway, Bellare, Vaudenay and other authors) one level of a cryptographic construction is modelled as a pseudo random functions (or permutations) generator, and is compared with an ideal (uniformly drawn) function or permutation generator with the same input and output sizes. Pseudorandomness results allowing to partially validate blockcipher features such as the so-called Feistel upper level structure of the DES construction, or to validate modes of operation of blockcipher such as for instance the CBC MAC mode were established. We believe that the use of such techniques will become more and more systematic to validate the structure of blockciphers or their modes of operation.

The above outline of the current status of cryptologic research on blockciphers could give the misleading impression that blockciphers design is now well ahead of blockciphers analysis, and one might wonder about the future of research on blockciphers after the recent selection of the Advanced Encryption Standard (AES). We believe however that the current research status of blockciphers is much less stable than might seem at first glance, and that blockciphers currently represent to many respects a less mature subfield of cryptography than public key cryptography. Examples of limitations of the current knowledge are the following:

- The state-of-the-art in security proofs for blockciphers is far from what has been achieved for their public-key counterparts. In particular, the security of existing blockciphers does not rely in a provable way upon the computational difficulty of well identified and well studied mathematical problems. Existing partial proofs of security for blockciphers are restricted to proofs of strength against restricted classes of attacks, and validation of some parts of their construction in the Luby and Rackoff paradigm. However, even the latter proof techniques are not systematically applied, as shown by the example of AES, which security is almost entirely empirical in nature.
- The only assertion one has for now concerning the security of AES is that for the time being, there exists no feasible shortcut attack. Since AES and several other recently proposed blockciphers make an extensive use of several algebraic structures (it has become common to use exponentials in a finite fields as the single non-linear component of a blockcipher), it cannot be precluded that further use of advanced algebraic techniques (Gröbner basis, probabilistic interpolation, quadratic approximations, genetic algorithms) might establish weaknesses in such ciphers. So new less statistical classes of attacks and their consequences on these ciphers are to be urgently investigated.
- Despite of the considerable number of public blockciphers now available, the AES and NESSIE competitions have revealed a considerable lack of diversity in the structure of currently proposed blockciphers.
- By analogy with the public-key world, it is now customary to view a blockcipher only as a building block to cryptographic functions (such as encryption or authentication). A mode of operation defines the usage of blockciphers to achieve cryptographic functions. The study of modes of operation and their partial validation in security models such as Luby and Rackoff's paradigm has not yet reached full maturity.

References

- [1] Data Encryption Standard (DES). FIPS 46-3, 1999.
- [2] J. Daemen and V. Rijmen. AES proposal : Rijndael, selected as the Advanced Encryption Standard (AES). FIPS 197, November 26, 2001, available from <http://www.nist.gov/aes>.
- [3] M. E. Hellman. A cryptanalytic time memory trade-off. *IEEE Trans. Infor. Theory* 26, 1980, pp. 401-406.
- [4] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Proc. Crypto '90*, LNCS 537, pp. 2-21, Springer Verlag, 1991.
- [5] M. Matsui. Linear cryptanalysis method for DES cipher. In *Proc. of Eurocrypt '93*, LNCS 765, pp. 386-397, Springer-Verlag, 1993.
- [6] M. Luby and C. Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Function. *SIAM Journal on Computing*, vol. 17, p.373, 1988.
- [7] K. Nyberg and L. R. Knudsen. Provable Security Against a Differential Attack. *Journal of Cryptology* 8(1), pp.27–37, 1995.
- [8] S. Vaudenay. Provable Security for Block Ciphers by Decorrelation. In *Proc. of STACS '98*, LNCS 1373, pp. 249-275, Springer-Verlag 1998.

4.1.2 Stream ciphers

Definition. By "stream cipher", we mean a symmetric algorithm producing a bit sequence (the keystream) of variable length, each bit of which depends on the secret key and the bit's position in the keystream; encryption of a message is done simply by bitwise XOR of the keystream to the plaintext. (More general definitions are possible, but the one given here is used in all practical applications.) Ideally, a stream cipher should give all the benefits of a long "one time pad", but with a short secret key.

This definition does not cover self-synchronising stream ciphers, which are treated in a subsection.

It is well known that stream ciphers can be constructed from block ciphers (or indeed from other cryptographic primitives), using modes such as Counter or Output Feedback Mode. In this section, we discuss dedicated stream ciphers, intended to have some advantage over the constructions from other primitives - usually that they can be faster, smaller, or both.

It is common to need to generate many different keystream sequences from the same secret key (these are sometimes called different frames of keystream). Stream ciphers typically use two inputs to generate keystream: a secret key, and an "initialisation variable" that is not generally secret. The initialisation variable changes from one frame to another, and it should be impossible for an attacker to find any useful relationship between keystream in one frame and keystream in another.

State-of-the-art. A small number of block ciphers are used for many different applications; DES and AES can be considered the old and new de facto standards. The same is not true of stream ciphers; where a dedicated stream cipher is required, new ciphers have generally been designed for each new application. Very often these have been in telecommunications equipment such as mobile telephones or radios, implemented in hardware to minimise power consumption; the radio interface encryption algorithms for GSM and GPRS are good examples. These have typically been subject to restricted usage undertakings, and often kept secret, although this is starting to change.

One algorithm that is used in a number of computing standards is RC4. This is a proprietary algorithm, not well suited for hardware but very simple indeed to implement in software. Although RC4 is not fundamentally broken, recent analysis has shown that some aspects of the key loading and initialisation are rather weak - and in particular that some ways of reusing RC4 over multiple frames, combining a common secret key with a varying frame counter, are easily broken.

A small number of other dedicated software stream ciphers (SEAL, SCREAM, SNOW) have been proposed. SEAL has been around for several years, but does not seem to be widely used, despite offering some advantages over block cipher based solutions. SCREAM and SNOW version 2 are very new.

One other stream cipher worthy of note is the shrinking (or self-shrinking) generator. This is not particularly efficient, but it is interesting in that it is probably the simplest of all known algorithms that still appear to be strongly resistant to cryptanalysis[2].

Cryptanalytic theory. Many stream ciphers intended for hardware implementation have been based on linear feedback shift registers, which can give guaranteed global statistical properties and long period. There are also several classes of internal state recovery attack against such stream ciphers: these include divide and conquer attacks, other correlation-based attacks, guess and determine attacks, and attacks based on multiple keystream sequences produced from a single secret key. A completely generic attack is the time/memory tradeoff.

Even ciphers that do not seem vulnerable to these internal state recovery attacks may be subject to a weaker attack: it may be possible to distinguish their keystream from a truly random sequence of bits with smaller effort than an exhaustive key search. This can lead to a genuine threat in some contexts, since it means that knowing some plaintext may allow the attacker to make deductions about other previously unknown plaintext. (It may be essentially irrelevant in other use contexts.) The significance of distinguishing attacks is the subject of some debate.

All the stream ciphers submitted to the NESSIE project proved to be more vulnerable than expected. And a distinguishing attack has been found against SEAL version 3 that defeats the cipher's design goals. There are really no dedicated stream ciphers that have yet withstood extensive public scrutiny without revealing at least slight weaknesses. The time is ripe for renewed research in this area.

Self-Synchronizing Stream Ciphers. In a (binary) synchronous stream cipher, a ciphertext sequence is produced from a plaintext sequence by bitwise XOR with the bits of a

keystream sequence; each keystream bit is a function of the initialisation variables (including a secret key) and the bit's position in the sequence. In a self-synchronising stream cipher, again a ciphertext sequence is produced from a plaintext sequence by bitwise XOR with the bits of a keystream sequence; but this time each keystream bit is a function of the initialisation variables (including a secret key) and the previous M bits of ciphertext, for some integer M .

A synchronous stream cipher is in a sense ideal for use in the presence of bit errors, because a single bit of ciphertext received in error will lead to just one bit of plaintext in error after decryption. However, if bit slippage (the loss or insertion of a bit) occurs, the result is catastrophic: all subsequent recovered plaintext is garbage. A self-synchronising stream cipher is for use precisely in situations where bit slippage is a possibility; the system recovers and decrypts correctly after any M bits of ciphertext have been received correctly, no matter what errors and slippages have gone before.

The only self-synchronising stream cipher that is in widespread use is the Cipher Feedback (CFB) mode of a block cipher. Papers have been published outlining more efficient design approaches [1, 2, 3], but no such concrete design has ever attracted significant public attention. It is unclear whether this is due to a lack of research interest in producing such ciphers or a lack of customer interest in using them.

References

- [1] J. Daemen. Cipher and Hash Function Design, Strategies based on linear and differential cryptanalysis. Ph. D. thesis, 1995.
- [2] J. Daemen, R. Govaerts and J. Vandewalle. On the Design of High Speed Self-Synchronizing Stream Ciphers. Singapore ICSS/ISITA '92 Conference, IEEE 1992, pp. 279-283.
- [3] U. M. Maurer. New approaches to the design of self-synchronizing stream ciphers. Advances in Cryptology - Eurocrypt '91, Springer-Verlag, 1991, pp. 458-471.

4.1.3 Pseudo-random number generation

Definition. A pseudorandom binary sequence is generated by an algorithm, with an input seed of bits that may be truly (although not necessarily uniformly) random. Since such algorithms are deterministic, the output of such an algorithmic process can never produce more random bits than were given as the input. For the purposes of this section we define a pseudorandom number generator (PRNG) to be a binary sequence generator whose output is not efficiently distinguishable from a truly random one using the means we have at hand. This is what would be required, say, for the generation of cryptographic keys.

The standard model for a pseudorandom number generator (PRNG) is as follows:

- Set $x_0 = \text{seed}$.
- Iterate $x_i = f(x_{i-1})$ for $i = 1, 2, \dots, n$.

- The output sequence is $b(x_1), b(x_2), \dots, b(x_n)$. $b(x_i)$ may be a single bit, or a number of bits, or an integer in a given range, or whatever.

The PRNG is defined by the state space $\{x_i\}$, the state transition function f , and the output function b .

If the input is a fixed length binary string, then the security requirements on a PRNG are essentially the same as the strongest requirement we might place on a stream cipher. So many applications for PRNGs are adequately addressed by stream ciphers (including constructions of stream ciphers from other cryptographic primitives, such as a block cipher run in counter mode). However, the requirements may be somewhat different:

- We may have lower expectations of the speed of a PRNG, but require stronger guarantees of security;
- We might have an ongoing slow stream of genuinely random input, rather than just a one-off input for initialisation. In this case the paradigm above would change; we have not just a single value "seed", but also successive values $\text{seed}(1)$, $\text{seed}(2)$, . . . that are additional inputs to the state transition function f . This model can be useful in cases where the amount of true randomness available at any instant is limited, but where the PRNG will be sampled at many instants - for instance, "date and time" may provide only limited entropy when sampled once, but the pooled entropy of the date and time of 20 successive samplings may be enough to resist a seed search attack[3].

In this section we concentrate on such specific requirements for PRNGs; we do not replicate arguments covered in sections on stream ciphers or other cryptographic primitives.

Theoretically-secure PRNGs. A sound theory of pseudorandomness emerged in the seminal works of Blum and Micali [1], and Yao [6] in the early 80's. In a theoretical sense the area was closed when it was shown in [4] that necessary and sufficient conditions for the existence of a pseudorandom generator is the existence of another fundamental primitive: the one-way function. This is a function easy to compute, but hard to invert. We do not know if such functions exist, but many strong candidates exist, such as cryptographic hash functions, or perhaps a good block cipher (the latter viewed as a function mapping keys to ciphertexts, keeping the plaintext fixed). It can also be noted that if one relaxes the requirement that the PRNG be efficiently computable, then secure (but completely impractical) PRNGs do exist that passes all polynomial-time tests.

Although the result mentioned [4] above gives an explicit construction of a pseudorandom generator from a one-way function, it is far too complex to have any practical implication. Key sizes of thousands (if not millions) of bits are needed to get any security out of the generator. This is due to the fact that one-wayness is in itself not a very strong property. In fact, a function may be hard to invert but still have some very undesirable properties. For instance, even if a function f is one-way, almost all of x may be easily deduced from $f(x)$. Secondly, generating a pseudorandom keystream will typically require iteration of some function and even if some f is one-way, it may lose its one-wayness if iterated. Thus, basing pseudorandomness on one-wayness alone is a delicate matter, appearing to require elaborate constructions.

However, if one assumes more than just one-wayness, *e.g.* that the function f is also a permutation, then the situation becomes much more favourable and much simpler constructions can be found. In fact, from the work of Blum and Micali [1] mentioned above, and later work by Goldreich and Levin [3], a general construction that is "almost practical" can be obtained. Basically, Blum and Micali [1] focus on the two undesirable properties mentioned above: information "leakage" of x through $f(x)$ and loss of one-wayness when iterated. They show that if f is a permutation and has at least a single bit of information, $b(x)$, that does not leak via $f(x)$, then a pseudorandom generator can be built. Goldreich and Levin [3] showed that every one-way function, in particular ones being permutations, have such a hard bit $b(x)$.

We thus have constructions for pseudorandom number generation that are provably secure under certain reasonable number theoretic assumptions. Examples for this are given by the Blum-Blum-Shub or the Blum-Micali generator. While Blum-Blum-Shub is based on iterative squaring modulo a product of two large primes, Blum-Micali is based on exponentiation modulo a single large prime number p . Their pseudorandomness is related to the problem of factoring a product of two large primes, and computing discrete logarithms modulo p , respectively. The Blum-Micali pseudorandom number generator can be proven to produce perfectly random bits to an observer that is limited to a probabilistic polynomial-time observation algorithm, assuming that the discrete logarithm problem can not be solved in probabilistic polynomial time. We can also note that the exact distributions of these generators have been studied, and are known to be statistically close to uniform in an absolute sense.

There are also constructions based on assumptions about symmetric primitives. For instance, if a block cipher is assumed to be a pseudorandom permutation, then running it in counter mode gives a provably strong PRNG [2]. However, the assumption that a block cipher is a pseudorandom permutation is quite a strong one. A weaker assumption, on a par with that made in the Blum-Micali construction, is that we have a one-way permutation - that is, a block cipher that is hard to invert. The BMGL construction [5] gives strong security bounds based on the related assumption that the underlying block cipher, when iterated, is hard to invert. We also note that the security analyses done in the above mentioned works do not take memory consumption into consideration, and therefore do not exclude time-memory trade-off attacks.

References

- [1] M. Blum and S. Micali. How to Generate Cryptographically Strong Sequences of Pseudorandom Bits, *SIAM Journal on Computing*, Vol 13, No 4, 1986, pp. 850-864.
- [2] M. Bellare, A. Desai, E. Jokipii and P. Rogaway. A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation, 38th IEEE FOCS, 1997.
- [3] O. Goldreich and L. A. Levin. A Hard Core Predicate for any One Way Function. 21st ACM STOC, 1989, pp. 25-32.
- [4] J. Håstad, R. Impagliazzo, L. A. Levin and M. Luby. Pseudo Random Number Generators from any One-way Function, *SIAM Journal on Computing*, Vol 28, 1999, pp. 1364-1396.

- [5] J. Håstad and M. Näslund. BMGL: Synchronous Key-stream Generator with Provable Security, NESSIE submission, www.cryptoneessie.org, 2000.
- [6] A. C. Yao. Theory and Applications of Trapdoor Functions, 23rd IEEE FOCS, 1982, pp. 80-91.

4.1.4 Pseudo-random functions

A family of pseudo random functions (PRF) map, under the influence of a key, n bits to m bits (usually $m \leq n$, as the other case can be thought of as a pseudo random generator) in such a way that the output is computationally indistinguishable from random m -bit strings. Note that when $m = n$ and the family consist of one-to-one mappings, we have an n -bit block cipher.

Some applications of PRFs: challenge response protocols (e.g. shared secret user identification), random strings with random-access property (in contrast to typical pseudo-random generator, the k th m -bit segment can be generated in time independent of k) e.g. for packet-data encryption, for cryptographic hashing and message authentication, etc.

Formal definitions of pseudo random functions were first given in [2]. In addition it was shown that PRF families exist if pseudo random generators do. In [3], it was shown by a 3-round Feistel-type construction that pseudo random functions imply the existence of pseudo random permutations. Both these results can be considered complexity theoretic, and in fact, very few explicit constructions exist. One example is, however, the number-theoretic construction of [4].

There are also some result on how to turn PRFs lacking certain properties into more useful ones. One result along these lines is the well-known HMAC construction, based on [1]. The results there shows how to turn an already existing PRF family with fixed input length into family allowing variable input length.

References

- [1] M. Bellare, R. Canetti, and H. Krawczyk: “Cascaded Pseudo-Randomness and its Concrete Security”, Proc. of 37th IEEE FOCS, 1996, pp. 504–513.
- [2] O. Goldreich, S. Goldwasser, and S. Micali: “How to Construct Random Functions”, J. ACM, vol 33, 1986, p. 792–807.
- [3] M. Luby and C. Rackoff: “How to Construct Pseudo Random Permutations from Pseudo Random Functions”, SIAM J. Comp., vol 17, 1988, p. 373–386.
- [4] M. Naor and O. Reingold: “Number Theoretic Construction of Pseudo Random Functions”, Proc. of 38th IEEE FOCS, 1997, pp. 458–467.

4.1.5 Cryptographic hash functions

Cryptographic hash functions play a fundamental role in modern cryptography. Hash functions map arbitrary length messages into short fixed length digests which after that can be used by other cryptographic primitives. For a good cryptographic hash function this digest is a "unique" representative of the original message in the sense that it is hard to forge different messages having the same digest or even having similar-looking digests. In contrast to other cryptographic primitives computation of a hash function does not depend on any secret information.

Hash functions provide cryptographically strong data integrity which is crucial in many applications, for example in digital signature schemes. This cryptographic hash functions a very important element in many cryptographic protocols. For example, instead of digitally signing lengthy messages it is enough to sign fixed length digest of a message. One of the main applications for hash functions is in compressing long messages into short fixed-length message digests which can be then signed by digital signature schemes.

Analysis of cryptographic hash functions is still a young area with many open problems. Similar to block-ciphers the natural division of problems is between design of custom compression functions which compress fixed length messages into fixed length outputs, adoption of existing block-cipher primitives in a hashing-mode, and between design of secure hashing schemes around secure compression functions primitives. There has been much progress in the design of provably secure hashing schemes in the last years. It seems however that we still lack a basic methodology for constructing sound custom compression functions. The last decade of research concentrated on construction of MD4-like functions (MD5, SHA, RIPEMD) and was characterized by a sequence of design-attack-redesign efforts, which lead to design of several efficient primitives with no obvious attacks found. As a future goal it is desirable to have more provable security properties for the designed primitives while retaining similar efficiency level. Another class of hash functions are those constructed from the block ciphers. These would benefit from progress which is made in the block cipher area. However it is still an open problem, what requirements for blockciphers are sufficient in order to produce a secure hash function. The opposite question is also of interest.

References

- [1] National Institute of Standards and Technology. The Secure Hash Algorithm (SHA-1). NIST FIPS PUB 180-1 *Secure Hash Standard*, U.S. Department of Commerce, April 1995.
- [2] H. Dobbertin, A. Bosselaers and B. Preneel. RIPEMD-160, a strengthened version of RIPEMD. In *Proc. of Fast Software Encryption '96*, pp. 71–82, LNCS 1039, Springer-Verlag, 1996.
- [3] I. Damgard. A design principle for hash functions. In *Proc. of Crypto '89*, LNCS 435, pp.416-427, Springer Verlag, 1990.
- [4] H. Dobbertin. Cryptanalysis of MD4. *Journal of Cryptology*, vol. 11, no. 4, pp. 253- 271, 1998.

- [5] F. Chabaud and A. Joux. Differential collisions in SHA-0. In *Proc. of Crypto '98*, Lecture Notes in Computer Science 1462, pp. 56-71, Springer-Verlag, 1998.

4.1.6 Message Authentication Codes

A special class of cryptographic hash functions which include user specified secret key into the hashing process are called Message Authentication Codes (MACs). Digests provided by MACs depend in an intricate way on both the original message and on the user specified key. It should be impossible to forge MACs without the knowledge of the secret key. The MACs, in addition to integrity applications, are used for data authentication and identification in symmetric key-schemes.

In the last years MACs have enjoyed much progress in generic provably secure MAC schemes resulting from appropriate modes of operation of hash functions, universal hash functions, or blockciphers (HMAC, UMAC, RMAC, etc.), but some research and standardisation is still needed in this area. Last of all, very few dedicated MAC designs have been proposed so far.

References

- [1] J.L Carter and M. N. Wegman. Universal hash functions. *JCSS*, 18(2), pp 143-154, 1979.
- [2] M. Bellare, R. Canetti and H. Krawczyk. Keying Hash Functions for Message Authentication. In *Proc. of Crypto '96*, LNCS 1109, Springer Verlag, 1996.
- [3] M. Bellare, J. Kilian and P. Rogaway. The security of the cipher block chaining message authentication code. In *Proc. of Crypto '94*, LNCS, Vol. 839, Springer-Verlag, 1994.
- [4] E. Petrank and C. Rackoff. CBC MAC for real-time data sources. *Journal of Cryptology*, Vol. 13, No. 3 (2000), pp. 315-338.

4.2 Asymmetric cryptography

4.2.1 Public key encryption

Provable security. At first glance, with the advent of standardization (RSA PKCS, ISO, IEEE P1363, CRYPTREC, NESSIE), one may think that the public-key encryption problem is solved: we now know several "secure" and efficient public-key encryption schemes. But the situation may be misleading: it is not clear at all that this is the end of the road, because everything depends on what is meant by "security".

The definition of security depends on two things: the goal of the adversary and the means of the adversary. Today, there is a consensus [2] that the goal of the adversary should be at least "distinguishability" (the adversary selects two different plaintexts; the challenger randomly encrypts one of the plaintexts; the adversary must guess which of the plaintexts has been encrypted), while the means of the adversary should be at most an adaptive

chosen-ciphertext attack (the adversary has access to a decryption oracle). However, this consensus may be subject to change.

Another issue lies in the realization of such a security notion. Today, a "relative" approach is used, by translating a security property into a computational assumption on the hardness of a well-defined and well-known computational problem such as factorization, the e -th root problem, discrete logarithm, *etc.* If the computational assumption holds, then the scheme satisfies the security notion. The main advantage of this approach is that we can clearly identify the security assumption of the scheme, while the main drawback is that we do not obtain any absolute proof of security: we merely have replaced a complex assumption in a complex world by a simpler assumption in a simpler world. The status of the computational assumption is subject to changes, and it is often difficult to compare different assumptions (for instance, comparing factorization with discrete logarithm). Besides, in most practical schemes known including those recommended in current standards, the translation (in the security analysis) from the "complex" world of schemes to the "simpler" world of computational problems is not meaningful (and sometimes, not even known to be correct) for the size of parameters currently in use. To tackle this issue, an idealization of the modelization of hash functions, the so-called random oracle model, has been popularized by Bellare and Rogaway (notably in [3]). In such a model, the hash functions used in encryption schemes are viewed as ideal random functions, which allows a simpler security analysis. However, the random oracle model is only an idealization: it is known not to be completely sound from a theoretical point of view, as there exist (theoretical) schemes which are provably secure in the random oracle model and still insecure for any choice of the hash function.

Alternatives to RSA. The RSA cryptosystem is the most widely used public-key encryption scheme. But this does not mean that RSA is the only public-key encryption known, or that we should not look for alternatives. In fact, finding alternatives to RSA has been a major topic of research since the appearance of public-key encryption. Several reasons explain the importance of this topic:

Security: It is good practice not to put all eggs in the same basket. In case factorization (or the e -th root problem) turns out to be much easier than expected, which we unfortunately cannot predict, one would like to have a ready-to-use alternative solution. Besides, we already know that factorization might become easy if quantum computers come to life.

Key-size: The RSA keys are getting bigger and bigger: while in 1977, it was conjectured that a 428-bit RSA modulus would provide adequate security, one is now able to factor an RSA modulus of bit-length 528 [1], and the minimal recommended bit-length is now at least 1024 (for long-term security, the recommended bit-length is much larger). Because the best factoring algorithm has subexponential complexity, doubling the security requires much more than adding a single bit to the key, as in symmetric cryptography. Interestingly, we now know other public-key encryption schemes with a significantly smaller key-size.

Efficiency: The increase in key-length has another implication: it makes the RSA scheme less and less efficient. Roughly speaking, in usual implementations of RSA, doubling the

bit-length of the RSA modulus slows down encryption and decryption by a multiplicative factor respectively of 4 and 8. Eventually, there might be a time when RSA keys become so big that the RSA scheme will no longer be practical.

The current alternatives to RSA can be roughly divided in two parts, depending on the type of strategy used:

- Shortening the key-size. This can be done by replacing the integer factorization problem with a potentially much harder problem. A major example is the discrete logarithm problem for algebraic varieties over finite fields: if the variety is well-chosen, no subexponential algorithm is known, which has led to the development of the so-called elliptic curve cryptography and its variants such as hyper elliptic curve cryptography, *etc.* In those schemes, the private key can be as small as 160 bits, and the underlying mathematics are arguably more involved than in RSA. However, more “complex” mathematics do not necessarily mean more security: for instance, it was shown that discrete logarithm for high-genus curves was much easier than for elliptic curves, and discrete logarithm for certain elliptic curves was not harder than discrete logarithm over finite fields. Another way of shortening the key-size is to provide a compact representation: this is the case in the XTR [8] or the LUC cryptosystems. XTR and LUC are simply discrete logarithm schemes in certain finite fields such that a compact representation of the elements is known. In those schemes, the key-length is a fraction of the RSA key-length: LUC achieves an improvement factor of 2 while XTR achieves an improvement factor of 3. It should be stressed that the best algorithm known to break such schemes is still subexponential, which makes any asymptotical improvement rather limited: the keys for XTR/LUC will grow much faster than for ECC keys. When the key-size of the scheme is less than that of RSA, the efficiency of the scheme is usually better, even though the basic operations may be more costly: in some sense, one is willing to trade many modular multiplications by a much smaller number of more complicated operations (like elliptic curve additions).
- Using more efficient operations than modular exponentiation. A popular strategy is based on complexity theory, more precisely on the so-called NP-hard problems. NP-hard problems are particular computational problems which are provably as hard as any problem of a large and natural class of computational problems: if an NP-hard problem can be solved efficiently (asymptotically speaking), then all the problems of the class can also be solved efficiently, which is considered unlikely. Thus, it is believed that NP-hard problems are hard, at least in the worst-case. It turns out that there are several NP-hard problems which only involve basic arithmetic. Cryptography has tried to use such problems to build cryptosystems, by transforming an easy instance into a potentially hard instance. The oldest example is the Merkle-Hellman cryptosystem based on the subset sum (also called knapsack) problem. Although the Merkle-Hellman scheme was broken shortly after its introduction, many schemes have since tried to apply the same principle, like:
 - NTRUEncrypt [6] and other lattice-based cryptosystems, which are based on the hardness of lattice problems. In NTRUEncrypt, the basic operation is a convolution product (polynomial multiplication modulo a certain polynomial). In other

lattice-based cryptosystems, the basic operation is the reduction of a vector modulo a lattice basis. Compared to other lattice-based cryptosystems, NTRUEncrypt is more efficient because it uses particular lattices for which a compact representation is known, which allows to significantly reduce the keysize.

- Coding-based schemes (McEliece/Niederreiter) where the basic operation is a multiplication by a binary matrix. The efficiency is similar to that of lattice-based cryptosystems (other than NTRUEncrypt), in terms of encryption/decryption rate and keysize.
- Schemes based on multivariate polynomial equations over a finite field. In such schemes, which are variants of the Matsumoto-Imai cryptosystem, the plaintext is viewed as a solution of a system of multivariate polynomial equations. Encryption and decryption can be done faster than RSA, but the keysize is similar to that of coding-based schemes: one needs to store the coefficients of the system of equations.

Generally speaking, all the previous schemes can encrypt and decrypt faster than RSA, but they require a larger keysize. One may add to the previous schemes another family based on non-commutative groups (which includes the braid groups), but at the moment, there seems to be serious security concerns for that family.

Special encryption. It turns out that in several applications, it is useful to have an encryption scheme with special properties, and not just a basic encryption scheme. One can distinguish several interesting features:

Homomorphic encryption 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. The Paillier cryptosystem [7] is the most efficient additive homomorphic scheme known in terms of bandwidth.

Identity-based encryption Key management is one of the biggest issues in public key cryptography. To guarantee the origin of public keys, one usually relies on certificates and a public key infrastructure. In an identity-based scheme, the public key infrastructure is much simpler: there is only a single authority. Any bitstring (such as the identity of the user) may be a public key: to obtain the corresponding private key, one must ask the authority to deliver the private key. The most efficient identity-based cryptosystem known is the recent Boneh-Franklin cryptosystem [5].

Traitor tracing The concept of a traitor tracing scheme, introduced by Chor, Fiat and Naor at Crypto '94, aims to discourage subscribers from giving away their private keys. One way to obtain a traitor tracing scheme is to build a special public key encryption scheme in which there is one public encryption key, but many private decryption keys. If some digital content is encrypted using the public key and distributed through a broadcast channel, then each legitimate user can decrypt using its own private key. However, if a coalition of users collude to create a new decryption key then the scheme has the additional property that there is an efficient algorithm to trace the new key to

its creators. The most efficient traitor tracing scheme known is a variant of the recent Boneh-Franklin traitor tracing scheme [4].

References

- [1] F. Bahr, J. Franke and T. Kleinjung. Factorization of the 158-digit cofactor of $2^{953} + 1$. Public announcement, January 19th 2002.
- [2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Proc. of Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, 1998.
- [3] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption. In *Proc. of Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, 1995.
- [4] D. Boneh and M. Franklin. An Efficient Public Key Traitor Tracing Scheme. *Advances in cryptology – Crypto '99*, Lect. Notes Comput. Sci. 1666, Springer-Verlag (1999).
- [5] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. *Advances in cryptology – Crypto '01*, Lect. Notes Comput. Sci. 2139, Springer-Verlag (2001), pp. 213–229.
- [6] J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: a Ring based Public Key Cryptosystem. In *Proc. of ANTS III*, LNCS 1423, pages 267–288. Springer-Verlag, 1998. First presented at the rump session of Crypto '96. New versions at www.ntru.com.
- [7] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology - Eurocrypt '99*, LNCS vol. 1592, Springer-Verlag, 1997, pages 223-238.
- [8] A. K. Lenstra and E. H. Verheul. The XTR public key system. In *Advances in Cryptology - Crypto '00*, LNCS vol. 1880, Springer-Verlag, 2000.

4.2.2 Digital signature

Provable security. As already mentioned in the public-key encryption part, the meaning of security is debatable. The situation is definitely worse for signature.

The definition of security depends on two things: the goal of the adversary and the means of the adversary, the information that is available to him. Everybody agrees that the main goal of signature is non-repudiation: a user who signed a message should not be able to later deny it. However, this notion is not precise enough for security analyses. The standard model to capture such a security notion is the so-called *resistance to existential forgeries*, where an adversary should not be able to produce an existential forgery, that is, a new valid message-signature pair, whatever the message is (even a meaningless message).

When one further considers the means of the adversary, things get even more complicated. The widely admitted scenario is the so-called *adaptive chosen-message attacks*, in which the adversary has unlimited access to a signing oracle which provides valid signatures for any

message adaptively chosen by the adversary. However, this gives rise to a rather ambiguous security notion, as an existential forgery may be interpreted in either of the following ways:

- a valid message-signature pair, for a new message;
- a new valid message-signature pair (for a new message, or a new signature for an already signed message).

The first one is in fact the definition of an *existential forgery*, while the second one is coined as *weak existential forgery*, or *malleability*. It is therefore not clear which security notion is the best one in practice: “no existential forgery” or “non-malleability”. Legal aspects have to be considered too.

The other issue about security proofs is related to the computational assumption on which relies the security of the signature. In other words, one needs to consider the hardness of a well-defined and well-known computational problem. Whereas trapdoor one-way functions (problems) are required for public-key encryption, one-way functions (problems) are sufficient for signatures. Therefore, many more problems can be used for signatures, and thus many more computational assumptions have been defined which make comparisons much more difficult. Paradoxically, whereas more problems can be used to build signatures than for encryption, there have been very few proposals for secure signatures. Very few efficient security reductions exist, even in idealized models, such as:

- the random oracle model, in which hash functions are viewed as ideal random functions;
- the generic model, in which a group structure is considered generic, meaning that the adversary has access to the internal group law through an oracle.

Alternatives to RSA. Like in encryption, RSA is the most widely used signature scheme. But this is not an ideal and perfect scheme, and we should look for alternatives, for similar reasons as for encryption:

Security: In case factorization (or the RSA problem) turns out to be much easier than expected.

Key-size: The RSA keys are getting larger.

Efficiency: The increase in key-length makes efficiency worse.

In contrast with encryption, many problems have been introduced to build signature schemes. Therefore, many alternatives to RSA exist, but very few are competitive from either the key/signature-size, efficiency or security. We can divide them in three parts:

- Shortening the key or signature sizes. This can be done by replacing integer factoring with the discrete logarithm problem, in finite fields or elliptic curves. In finite fields, the discrete logarithm admits sub-exponential algorithm, but the keys (at least the signing key) may be shorter. Furthermore, the signature can be reduced down to 240 bits (to

be compared with 1024 bits with RSA). If one uses well-chosen elliptic curves, both the signing and the verification keys can be as small as 160 bits.

Other groups, in which the discrete logarithm problem seems hard, have also been proposed, with more or less compact representation of elements. This may also improve the key size, but the size of the signature hardly becomes less than 240 bits.

- Using more efficient operations than modular exponentiation. As already explained in the public-key encryption part, several NP-hard problems from complexity theory have been used in cryptographic schemes, in order to achieve better efficiency. Modular exponentiations (in RSA, or the discrete logarithm setting) are quite costly, while hard problems, involving basic arithmetic only, do exist. But in order to obtain a really efficient signature scheme, a trapdoor is often required: the security does not exactly rely on the intractability of recovering the signing key, but the signing key can be seen as a trapdoor which converts a hard instance (the public key) into an easy one. This is the case in recent proposals:
 - NTRUSign [1] and other lattice-based signatures, which are based on the hardness of lattice problems. The signing key provides a good basis of the lattice, which can be used to approximate closest vectors in the lattice. However, there are major security concerns: It has recently been shown that in such schemes, each signature leaks information on the private key. Once enough signatures are published, the lattice reduction problem can be reduced to a simpler lattice problem, whose complexity has yet to be well-understood.
 - Coding-based schemes (a recent variant [2] of McEliece/Niederreiter cryptosystems), which are based on the hardness of decoding random linear codes. The signing key converts the randomly looking code into a well-known Goppa code, easy to decode.
 - Schemes based on multivariate polynomial equations over a finite field. Such a system of equations is difficult in general, but the signing key helps to convert it into a system easy to solve.

As already remarked in the public-key encryption part, all these schemes are very efficient, but they require a larger key-size. Furthermore, even if they are based on NP-hard problems, their security is not equivalent to the underlying problem. Thus, the trapdoor may weaken the problem.

- Using NP-hard problems, without any trapdoor. The NP-hardness of a problem does not guarantee that a problem is really hard: it only gives strong evidence that there exist hard instances which cannot be solved within polynomial time. When one introduces a trapdoor when defining an instance, the chance to have a really hard instance gets smaller. More than 15 years ago, techniques were developed to allow someone, who knows a solution to a problem, to prove his knowledge without revealing anything else: zero-knowledge proofs of knowledge. Such proofs can be used for signing: the public key is a hard instance, while the signing key is a solution; a signature is a zero-knowledge proof of the knowledge of the solution. However, zero-knowledge proofs are only possible interactively, and signatures should depend on the message. The Fiat-Shamir paradigm solves both problems at once: the verifier is replaced by a hash function on the message,

and a few additional data. This paradigm converts any interactive zero-knowledge proof of knowledge into a signature scheme, provably secure in the random oracle model. The security level is exactly the intractability of the underlying NP-hard problem. PKP (Permuted Kernel Problem) proposed by Shamir has been the first efficient interactive zero-knowledge proof of knowledge using a NP-hard problem, which is indeed hard for reasonable sizes. A few other proposals followed, but NP-hard problems which are intractable for small parameters are not numerous. Anyway, these signature schemes are not really realistic in practice: they show large signatures, and rather inefficient algorithms for signing and verifying. But they would be the unique alternatives in case problems from number theory and trapdoor NP-hard problems become easy (which might come from quantum computing).

Special signatures. It turns out that in several applications, it is useful to have a signature scheme with additional properties. One can distinguish several interesting features:

Proxy signature. Delegation of signature is an important feature (*e.g.* in GSI, the security infrastructure of the Data Grid). With any signature, it is possible to achieve delegation, using certificates. But this entails larger signatures, and verification gets slower. DL-based proxy signatures have already been proposed to avoid these drawbacks.

Designated-verifier signature. The main goal of signature is non-repudiation. However, one may want to convince someone that he is the sender, but in such a way that this signature cannot be revealed to anybody else.

Undeniable signature. In the same vein as designated-verifier signatures, undeniable signatures protect the signer against publication of the signature, because the signer has to be present for confirming (or denying) a signature. However, the recipient is sure that denial is impossible for a valid signature.

Blind signature. This is an essential tool for anonymity. Such a signature is the analogous of homomorphic encryption. Indeed, it allows someone to apply a transformation on a message (usually applying a group law) so that a signature on such a blinded message helps him get a valid signature on the initial message. The signer eventually has no information about the message and the signature either. This is widely used for electronic cash and electronic voting, where anonymity is a critical issue.

Ring/group signature. This is another ingredient to protect anonymity. Ring or group signatures allow someone to sign in the name of a group without revealing more about his identity. The difference between ring and group signatures is that, in the former the group can be defined on the fly by the user when he wants to sign with perfect anonymity, whereas in the latter, a group manager has the ability to revoke anonymity.

Short signature. In several applications, efficiency and key size may not be that much important, but the size of the signature is. There is no theoretical objection against signatures as short as 80 bits. Some proposals have already been put forward, but none is really convincing.

Identity-based signature. As already noticed, key management is one of the biggest issues in public key cryptography. To guarantee the origin of public keys, one usually relies

on certificates and a public key infrastructure. In an identity-based scheme, the public key infrastructure is much simpler: any bit-string (such as the identity of the user) may be a public key. Identity-based signatures exist for a long time, but based on integer factoring only.

References

- [1] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. Silverman and W. Whyte. NTRUSign: Digital Signatures Using the NTRU Lattice. To appear in Proc. of CT-RSA '03, LNCS, Springer-Verlag.
- [2] N. Courtois, M. Finiasz and N. Sendrier. How to Achieve a McEliece-Based Digital Signature Scheme. In Proc. of Asiacrypt '01, LNCS, Springer-Verlag.

4.3 Implementation aspects and realization

4.3.1 Flexibility

For real-life applications one needs to implement cryptographic protocols in hard- or software. In addition to the potential weaknesses in the actual algorithm or protocol which are considered above, the realization itself might lead to further attacks. Prime examples for those can be found in the rapidly growing field of side channel attacks, including power and RF radiation analysis attacks.

The other important area in which trends will be identified are efficient implementation techniques. The predicted importance of embedded applications, often in constraint environments, adds special importance to this field.

Obviously, if one wants to use cryptographic techniques in any kind of application, the cryptographic mechanisms must be implemented, i.e., realized, on some kind of physical platform. An example is a digital signature algorithm realized on a smart card processor. Roughly speaking, efficient implementation can be defined as "fast algorithms in hardware and software." However, a much more detailed look is necessary in order to solve the problems at hand.

Initially it is important to distinguish between the different implementation platforms which are relevant for cryptographic applications. We distinguish between three types of software and hardware platforms.

1. Software implementations using general purpose hardware
2. Implementations using dedicated hardware (ASIC, FPGA)
3. Embedded software implementations (such as smart cards)

A major issue when practical cryptographic implementations are considered is the fact that the computations start leaking information, whereas the mathematical algorithm doesn't take

this problem into account. For example, side-channel analysis is a form of attack against secure tokens by which secret data is extracted without damaging the device itself. By monitoring the execution time, the power consumption or the electromagnetic radiation of an Integrated Circuit, it is frequently possible to infer information about the processed data. We address this topic in other sections of this document.

References

- [1] W. Rankl and W. Effing. Smart Card Handbook. 2nd edition, New York, John Wiley & Sons, 2000.
- [2] P. Montgomery. Modular multiplication without trial division. In *Mathematics of Computation*, vol. 44, April 1985.
- [3] E. Brickell, D. Gordon, K. McCurley and D. Wilson. Fast exponentiation with precomputation. In *Proc. of Eurocrypt '92*, Lecture Notes in Computer Science volume 658, Springer Verlag, 1993.
- [4] H. Sedlak. The RSA cryptography processor. In *Proc. of Eurocrypt '87*, Lecture Notes in Computer Science, Springer Verlag, 1987.
- [5] C. Ko, T. Acar and B.S. Kaliski Jr. Analyzing and comparing Montgomery multiplication algorithms. *IEEE Micro*, vol. 16, 1996.

4.3.2 Efficient hardware

There are (traditional) stand-alone hardware implementations of cryptographic algorithms using different hardware environments such as application specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs).

ASIC implementations are characterized by increasingly high performance and gate/functional density, relatively low per-unit costs, relatively high initial design costs (NRE — non-recurring engineering costs) and design time. In addition, ASICs have the drawback that functional changes in fielded devices are hard to perform.

There is also reconfigurable hardware which means in practice very often field programmable gate arrays, or FPGAs. FPGAs are pieces of hardware which can be programmed, that is, the FPGA functionality is not fixed. Current commercial FPGAs allow designs with a complexity equivalent to a few 100,000 gates, which is sufficient even for very complex cryptographic applications such as 2048 bit RSA. Compared to ASICs, FPGAs have the advantage of relatively low initial design costs and time. Another advantage is that designs can be altered, both during prototyping and even in fielded applications, assuming the infrastructure needed for reprogramming is provided in the application. The drawbacks are relatively high costs per unit and higher power consumption compared to ASICs.

References

- [1] J. Goodman and A. Chandrasekaran. An energy efficient reconfigurable public-key cryptography processor +architecture. In *Proc. of CHES '00*, pp. 174-191, LNCS 1965, Springer-Verlag, 2000.
- [2] G. Orlando and C. Paar. A High-Performance Reconfigurable Elliptic Curve Processor for $+GF(2^m)$. In *Proc. of CHES '00*, pp. 363-378, LNCS 1965, Springer-Verlag, 2000.

4.3.3 Efficient software

First, there are software implementations on (traditional) computers, such as PCs/laptops and workstations. Such implementations are characterized by relatively high processor performance (e.g., memory-rich Intel processors clocked in the giga Hertz range), and software development in a high-level language such as C++ or Java. The development environments are often relatively mature and user friendly.

Many crypto schemes, most notably public-key schemes, are computationally very expensive. Prime examples are schemes such as RSA, Diffie-Hellman key exchange, or the Digital Signature Algorithm (DSA), all of which require currently arithmetic with 1024-2048 bit operands. It is quite realistic that 3072-4096 bits will be needed in the not-so-distant future. It is not hard to imagine that arithmetic involving operands with those bit lengths can be very slow unless they are implemented with care. However, current PCs and workstations have - due to Moore's law - become sufficiently fast to execute even complex public-key algorithms in a time that is fully sufficient, say, below 100msec, on the client side of a network connection. In contrast, for a server which has to communicate with many clients during a given time interval, e.g., the server of an e-commerce site, computationally intensive cryptographic operations are still a problem. With the expected increase in secure Internet communications, for instance due to the increasing popularity of IPsec, servers with high cryptographic load will become more commonplace. Also, since the data transmission rate has been growing exponentially in the last decade or so (at an even higher growth rate than VLSI technology), there is a need for extremely high-speed implementation for encryption functions taking place at the network level. In either case, advances in the area of high-speed software and hardware algorithms are needed in order to provide adequate security in the future.

4.3.4 Embedded software

There are embedded software implementations. Examples are security protocols running on a PalmPilot PDA or a radio frequency ID tag which might be used as a smart luggage tag in airports in the future. Such applications are usually based on constrained processors. Very roughly speaking, the micro processors found here may range from those with 32 bit clocked at 100MHz down to 8 bit models with a 5 MHz clock. In addition, also memory is constrained. Those processors are often programmed in more machine-oriented languages, for instance assembly or low-level C. It is not uncommon that proprietary languages and developments tools must be used. The software written for embedded applications is sometimes referred to as "firmware". On the other hand, another important step of the development of an embedded system is the design of cryptographic coprocessors which are used to accelerate typical

integrated chip micro-controllers. A typical example of such embedded implementations are those used for smart card technologies.

A smart card is a plastic card containing an embedded tamper-resistant integrated circuit (IC). A smart card IC typically consists of an 8-bit (up to 32-bit) micro-controller equipped with a few tens of kilobytes of Read- Only Memory (ROM) which contains the operating system, a few tens of kilobytes of electrically erasable programmable read-only memory (EEPROM) which holds secret keys and file access codes, and about one kilobyte of random-access memory (RAM) in which cryptographic computations can be performed. Thus typical public key computations such as modular exponentiation of 1024-bit numbers are rather inconceivable on this kind of chip. Even secret key cryptographic computations sometimes become a bottleneck for an application using a smart card.

The purpose of a smart card is to provide secure storage of sensitive data and applications. Highly sensitive data is never communicated outside of the card; all operations are carried out by the operating system inside the card. The operating system also handles security and data access for each of the applications in the smart card. This way a smart card is considered as a tamper-resistant device. As such, all sensitive computations involving secret keys or sensitive data have to be performed without the keys ever being leaked to the outside world. All cryptographic operations such as encryption, digital signature generation or message authentication code computations have to be performed in a highly constrained environment.

Smart-card techniques for public key cryptography or secret key cryptography have been developed to overcome this technical difficulty. A whole range of new algorithms have been designed to specifically be able to handle large integer computations using as few memory bytes as possible. For instance, specific cryptographic coprocessors which provide modular exponentiation over 1024-bit integers have been invented and embedded into the design of smart card micro-controllers. Hardware accelerators also start to appear for secret key cryptographic computations (DES accelerators or SHA-1 accelerators) as well as combined DES and Elliptic Curve (over binary fields) coprocessors.

Latest smart card micro-controllers include public key coprocessors such as the MAP and SuperMAP (Modular Arithmetic Processor) designed by Fortress and provided by ST and NEC, the ACE (Advanced Cryptographic Engine) and Crypto2000 provided by Infineon or the FameX and SmartXA families provided by Philips. Other smaller scale designs handling 512 bit moduli have appeared in the past, but nowadays 1024 bit and even larger moduli can be handled without too much concern in high security applications.

These cryptographic coprocessors rely on specific algorithms enabling fast modular multiplication such as the well-known Montgomery Reduction or the Sedlak algorithm, Barret's Reduction Method or the Quisquater-Couvreur technique. All these methods decompose large integer computations over smaller registers of t bits, t being a power of 2, in the fastest possible ways using as few memory as possible. Several different tradeoffs are achieved. Recently other computation techniques such as size-doubling for modular arithmetic have been proposed to further speed-up public key cryptography in constrained environments. Other methods such as point-halving or sliding window methods apply to specific public key cryptosystems based on elliptic curves and also found an application for optimizing smart card implementations.

All smart card and chip manufacturers rely on these techniques today to propose their latest public key technology enabled products. Users include application developers and smart card issuers providing digital signature facilities and promoting the use of smart cards in public key infrastructures. Examples of such public key enabled devices include the German Geldkarte or the WAP Identity Module (WIM) for secure signature generation. On a very large scale, cryptographic smart cards are used wherever security is essential for privacy or financial purposes. Secure systems relying on secret key cryptography include the Global System for mobile communications (GSM) relying on the SIM card (Subscriber Identification Module), the Europay-Mastercard and Visa (EMV) system for financial transactions relying on a smart debit/credit card, national identity and healthcare programs using message authentication techniques and encryption relying on a personal smart card.

4.4 Dedicated attack techniques

4.4.1 Side-channel attack of an implementation

Performing a Side-Channel analysis on a secure token requires a sound knowledge in electronics, cryptography, signal processing and statistics. A now well known class of attack in this group is based on power consumption analysis of the device : Differential Power Analysis (DPA) and Simple Power Analysis (SPA), but also on timing analysis.

The concept of SPA consists in observing the variations in the global power consumption of the chip and retrieving from it some information that can help to identify a secret. For example, an increase in power consumption might indicate where a modular exponentiation is performed. In general, a SPA will give better results if the hardware architecture of the tamper-resistant device is known. DPA is more sophisticated than SPA: it consists in performing statistical analysis and correlation analysis on power consumption curves obtained from several executions of the same algorithm with different inputs to retrieve the secret information.

Timing attacks were a main issue in the past because several optimizations implied algorithms with varying timings depending on the data and/or the cryptographic keys in use. All current cryptographic algorithm implementations have to be designed with constant timing or should at least not depend on intermediate data and secret keys.

A more recent attack is Electromagnetic Analysis: it is based on the same techniques used for DPA and SPA, but the measured physical quantities are different. In this case, it is the Radio Frequency signals that are interesting. While also being a side-channel attack, Electromagnetic attacks differ in a number of points from power attacks. Since any electrical current flowing through a conductor induces electromagnetic (EM) emanations, it seems natural to look for the same phenomenon in the vicinity of a semiconductor. As the power consumption of a tamper-resistant device varies while data are being processed, so does the EM field and one may legitimately expect to extract secret information from a relevant EM analysis. However, this requires the design of special probes and the development of advanced measurement methods that focus very accurately selected points of the chip. EM's advantage is definitely its capability of exploiting local information. This geometrical degree of freedom is useful as it allows pinpointing the problematic spots that leak information. Power attacks' major advantage is undoubtedly the relative simplicity of electric measurements as opposed

to EM ones.

Several papers have been published on this topic over the last few years including attacks on block ciphers such as AES, DES, RC5, IDEA and others, attacks on hash functions such as SHA-1 and last but not least on public key cryptosystems such as RSA, DSA or Elliptic curve based algorithms. There have also been several new research results on protective countermeasures against such attacks and most of the recent papers in the area propose theoretical approaches to solve the issue of side-channel attacks.

The first method discovered right after the introduction of timing attacks was the so-called operation-constant implementation which basically makes sure the way the algorithm is implemented does not depend in any way on the individual values of the data being manipulated. The next concepts introduced after SPA attacks made their way to the public were current scrambling techniques and wait state introduction. These would disable the adversary from identifying the actual secret data manipulations (in time and absolute value) from the power measurements he could make.

The most recent countermeasures include data masking techniques such as boolean masking and arithmetic masking which ensure that, for each new execution, independent copies of the key and intermediate values are used during the cryptographic computation. This way first order correlations can be avoided in differential power consumption or electromagnetic radiation measurements. Equivalently, randomization of public key exponents, basis or moduli, or even randomized addition-subtraction chains may be used for elliptic curve cryptosystems to achieve decorrelation between the manipulated data and the actual secrets.

However more complex side-channel attacks have appeared which seem to bypass all previous known protective techniques, therefore close collaboration between cryptographers and engineers familiar with the issue of tamper-resistance is needed in order to refine future requirements for cryptographic algorithms

References

- [1] P. C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Proc. of Crypto '96*, Lecture Notes in Computer Science, volume 1109, Springer Verlag, 1996.
- [2] D. Boneh, R. DeMillo and R. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In *Proc. of Eurocrypt '97*, Lecture Notes in Computer Science, volume 1233, Springer Verlag, 1997.
- [3] J. Kelsey, B. Schneier, D. Wagner and C. Hall. Side Channel Cryptanalysis of Product Ciphers. In *Proc. of ESORICS '98*, Lecture Notes in Computer Science, volume 1485, Springer Verlag, 1998.
- [4] P. Kocher, J. Jaffe and B. Jun. Differential Power Analysis. In *Proc. of Crypto '99*, Lecture Notes in Computer Science, volume 1666, Springer Verlag, 1999.
- [5] K. Gandolfi, C. Moutrel and F. Olivier. Electromagnetic analysis: concrete results. In *Proc. of CHES '01*, Lecture Notes in Computer Science, volume 2162, Springer Verlag, 2001.

- [6] J.-J. Quisquater and D. Samyde. ElectroMagnetic Analysis (EMA) Measures and Counter-Measures for Smart Cards. In *E-Smart Smartcard Programming and Security*, Lecture Notes in Computer Science, volume 2140, Springer Verlag, 2001.
- [7] T. S. Messerges, E. A. Dabbish and R. H. Sloan. Investigations of power analysis attacks on smart cards. In *Proc. of CHES '99*, LNCS 1717, Springer-Verlag, 1999.
- [8] T. S. Messerges, E. A. Dabbish and R. H. Sloan. Power Analysis attacks of modular exponentiation in smart cards. In *USENIX '99*.
- [9] J. S. Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In *Proc. of CHES '99*, LNCS 1717, Springer-Verlag, 1999.

4.4.2 Active Attacks on an Implementation: Fault Induction

While side-channel attacks exploit additional information available during *regular* operations of a cryptographic device, an attacker can also try to induce faults into key-dependent computations and cause malfunctions in a cryptographic device. This can, e. g., be realized through suitable physical means. A recent example for this kind of attack is provided by the optical fault induction attacks against smartcards described in [3].

In particular, such ‘active’ attacks are outside the framework that is usually considered in the context of provable security. As discussed in [1], the additional operations needed for deriving a cryptographic scheme with provable security properties from a cryptographic primitive, can in fact introduce new vulnerabilities with respect to attacks based on fault induction. A similar problem is addressed in [2]: here a setting is considered, where an attacker is able to enforce a memory dump while a decryption takes place—therewith revealing potentially valuable information. However, the detailed interplay of theoretical (provable) security properties and such implementation-dependent attacks is not well understood yet, and still needs further exploration.

References

- [1] Marc Joye, Jean-Jacques Quisquater, Sung-Ming Yen, and Moti Yung. Observability Analysis — Detecting When Improved Cryptosystems Fail —. In Bart Preneel, editor, *Topics in Cryptology — CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 17–29. Springer, 2002.
- [2] Seungjoo Kim, Jung Hee Cheon, Marc Joye, Seongan Lim, Masahiro Mambo, Dongho Won, and Yuliang Zheng. Strong Adaptive Chosen-Ciphertext Attacks with Memory Dump (or: The Importance of the Order of Decryption and Validation). In Bahram Honary, editor, *Proceedings of Cryptography and Coding 2001*, volume 2260 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2001.
- [3] Sergei P. Skorobogatov and Ross J. Anderson. Optical Fault Induction Attacks. In Burton S. Kaliski Jr., Çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 2–12. Springer, 2003.

4.4.3 Alternative computing devices

Quantum computing. With the development of quantum information theory, it became clear that physical principles play a role in information theory in general and in cryptography in particular [9]. One of the remarkable properties of Quantum Information Theory is the no-cloning principle for qubits. Although this seems an undesirable property, Wiesner and Bennett [2] showed how this principle can be applied to build secure crypto systems. By encoding bits in non-orthogonal photon states, a first unconditionally secure Key Distribution System was developed in [2]. Nowadays this scheme is known as BB84. This protocol solves an important key-management problem in cryptography. Two users who want to communicate securely, can run the BB84 protocol to set-up a secret (classical) key. This key can then be used for encryption and authentication purposes during the communication. The security of this protocol has been shown in several papers as for instance in [3]. BB84 has given rise to the development of many other key distribution protocols and has inspired many researchers all over the world to investigate other applications of quantum mechanics in security. We first mention some research results on alternative Quantum Key Distribution (QKD) protocols. Bennett [9] developed a two state Quantum Key Distribution protocol: B92. An important practical disadvantage of the above-mentioned implementations of Quantum Key Distribution (BB84, B92) is that single photon sources are needed which are very difficult to build. Alternatives based on Squeezed states [8] and Coherent state light beams [6] have been proposed. The last proposal has the potential to bring Key Distribution protocols based on Quantum Mechanical Principles much closer to practicality.

Secondly, we provide a brief (not at all exhaustive) overview of many other quantum mechanics based security protocols, that have been developed. As an example we mention Quantum Secret Sharing Schemes [5, 10], Quantum Authentication Codes [1], Quantum Digital Signatures [5], Quantum Multiparty Computation [4] and the Hiding of bits in Quantum (Bell) states [11]. It was shown that many of these protocols have different properties from their classical counterparts. In quantum secret sharing for instance, it is impossible to share a qubit amongst $2n$ players such that any subgroup of n players is able to reconstruct the secret. This is an easy consequence of the no-cloning principle. Quantum Digital Signatures are based on the notion of Weak Quantum One Way Functions (WQOWF) [5]. On the other hand also some impossibility results have been proven. The most famous ones being the impossibility of Quantum Oblivious Transfer and Quantum Bit Commitment [7].

Another powerful application of quantum mechanics is Quantum Computing. It was shown by P. Shor that a quantum computer can factorise large numbers much faster than a traditional computer by making use of the superposition principle [9]. The core of this technique is based on Quantum Fourier Transforms (QFT) which allow to solve the so-called order finding problem, which is equivalent to the factoring problem. It turns out that an integer n consisting of L bits can be factored in $\mathcal{O}(L^3)$ operations [9]. Furthermore, it was shown that the discrete logarithm problem can be formulated in terms of the period finding problem which can also be solved efficiently on a quantum computer. In the summer of 2002, Hallgren proved that also so-called Pell's problem can be solved in polynomial time by a quantum computer. Summarised, the above mentioned applications of quantum computing imply that the potential existence of a quantum computer means a serious threat for most of nowadays public key crypto systems.

References

- [1] H. Barnum, C. Crepeau, D. Gottesman, A. Smith, A. Tapp: *Authentication of quantum messages*, quant-ph/0205128.
- [2] C.H. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin, *J. Cryptology*, **5**, 1992.
- [3] E. Biham, M. Boyer, P. O. Boykin, T. Mor, V. Roychowdhury: *A proof of the security of Quantum Key Distribution*, quant-ph/9912053.
- [4] C. Crepeau, D. Gottesmann, A. Smith, *Quantum Multiparty Computation*, STOC 2002.
- [5] D. Gottesman, I.L. Chuang: *Quantum Digital Signatures*, quant-ph/01050032.
- [6] F. Grosshans and P. Grangier: *Continuous variable Quantum Cryptography using Coherent States*, quant-ph/0109084.
- [7] D. Mayers: *The trouble with quantum bit commitment*, quant-ph/9603015.
- [8] D. Gottesmann, J. Preskill: *Secure Quantum Key Distribution using Squeezed States*, quant-ph/0008046.
- [9] M. Nielsen, I. Chuang: *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
- [10] A. Smith, *Quantum Secret Sharing for General Access Structures*, quant-ph/0011042.
- [11] B. Terhal, D. DiVincenzo, D.W; Leung: *Hiding Bits in Bell States*, quant-ph/0001087.

Special-Purpose Cryptanalysis Hardware.

Dedicated hardware for attacks on symmetric key algorithms. Special purpose hardware for cryptanalysis has been built at least as early as the 1940s, during World War II. In particular, the efforts by the British and US intelligent services to break German and Japanese ciphers using special purpose electro-mechanical devices have been well documented. Even though it appears almost certain that many other machines have been built by the intelligence community since then, there are no confirmed reports.

Among the first public attempts to use special-built hardware against modern ciphers was a proposal by Diffie and Hellman in 1977 for an exhaustive key search machine for DES. A very detailed and, in hindsight, accurate proposal for an exhaustive key search machine against DES was provided by Mike Wiener in 1993. The machine was estimated to cost \$1,000,000 and would find a key in 3.5 hours on average. In the report it was estimated that a key search machine could be built in about ten months by three people [9]. It should be stressed that both DES key search machines mentioned thus far were designs only, i.e., they had not been actually built.

This situation changed in 1998 with the EFF DES key search machine, designed by Cryptography Research, Advanced Wireless Technologies, and the Electronic Frontier Foundation (EFF). The machine was designed and built, and can test more than 92 billion keys per

second with over 37,000 search units. With use of this machine, the RSA DES Challenge on July 15, 1998 was won after searching for 56 hours. The DES Key Search Machine uses a sieve-and-check search process that can find keys even when little is known about the plaintext. Each chip processes two separate ciphertexts and contains a 256-bit vector specifying which bytes can appear in the plaintext — making it possible, for example, to find a key if the input message is simply known to consist of ASCII text. As assembled, the machine is housed in six recycled SUN-2 cabinets and consists of 27 circuit boards that hold over 1800 custom chips. Each chip contains 24 search units, which independently scan through a range of keys, filtering out those that do not pass the search criteria for both of the ciphertexts. Cost and development time were major factors in the machine's design. Most of the expenses were one-time research and design costs. The total project budget remained under \$250,000 [3].

At CHES 1999, Ivan Hamer and Paul Cho [5] presented a DES-cracking hardware on a field-programmable system called the Transmogripher 2a. The authors partially implemented a system consisting of four FPGAs and claimed that a fully implemented system will be able to search the entire key space in 1040 days. A fully implemented Transmogripher consists of 32 Altera 10K100 FPGAs and will be able to check $2^{29.6}$ keys per second at a clock rate of 25MHz. According to an estimate of the authors, a performance increase by a factor of eight could be reached, spending the same amount than the EFF for the DES Challenge III.

More recently, Michael Bond and Richard Clayton presented an implementation of a low cost DES-Cracker based on an Altera 20K200 FPGA kit for \$995 [2]. The central idea was to attack multiple keys in parallel. The same plaintext is encrypted under each of the multiple keys to get a test vector. The attack was performed by trying all keys in sequence but check for a match against any test vector value (check is faster than encrypt). Thus, for a typical case, a 2^{56} search for one key becomes a 2^{42} search for 2^{14} keys. With 2^{25} keys per second, a full key search would take on average about 68 years on a single machine, 25.4 hours with 16,000 machines.

As a conclusion of the development of DES attacks, a key length of 80bits and more is suggested for the use in symmetric cryptographic applications in the next 10 years.

Dedicated hardware for attacks on asymmetric key algorithms. Adi Shamir's TWINKLE (The Weizmann INstitute Key Locating Engine) device was first introduced at the rump session of Eurocrypt '99, with the full version of the paper being published at CHES '99 [8]. TWINKLE uses LEDs and opto-detectors to achieve free-space processing. Shamir estimated that the TWINKLE device would require \$1,000,000 to develop and cost \$5,000 per device. He estimates that TWINKLE could analyze 100,000,000 large integers, and find which integers completely factor over a prime base consisting of 200,000 prime numbers, in less than 10 milliseconds while operating at 10GHz. The development path for the VLSI is assumed to be short.

A major drawback of the TWINKLE device is the fact that it relies on the use of expensive Gallium Arsenide technology. Furthermore, there are several other unsolved technical problems concerning the realization of the design. Even though many ideas of TWINKLE are very innovative and interesting, there have been no known implementations of it.

Hea Joung Kim and William Mangione-Smith described in their FPGA2000 [6] paper how

to map the sieving process of the multiple polynomial quadratic sieve (MPQS) to a runtime reconfigurable and adaptive computing architecture build of FPGAs (called the Mojave configurable computing architecture). The goal was to combine offline optimization with runtime hardware reconfiguration in order to achieve higher performance than possible with either a general-purpose processor or a custom ASIC. The actual implementation was done on a single FPGA only, clocked at 16 MHz. The projected system with four devices achieves a speedup factor of 28 over Ultrasparc Workstations. The time that it takes to break RSA-129 is estimated to be two months. The use of this hardware for sieving with the number field sieve (NFS) is assumed to be significantly faster than MPQS. The authors claimed to be able to reduce the sieving time for the RSA-155 from 6 to 7 months down to weeks. Again, full-size versions of the attack hardware have not been built.

The proposal by Bernstein, presented at Eurocrypt'01 [1], suggested a circuit-based implementation of the matrix step of the number field sieve factorization algorithm. These circuits offer an asymptotic cost reduction under the measure “construction cost \times run time”. The matrix step is based on Schimmler's sorting algorithm which can be extensively parallelized. Bernstein suggests to replace sieving with direct smoothing testing with the elliptic curve method (ECM), which is asymptotically faster. According to Bernstein, special circuits built according to his proposal could factor integers that are 3.01 time longer than those factored by current hard- and software implementations. An analysis of Bernstein's factorization circuit done by Lenstra, Shamir, Tomlinson and Tromer [7] results in a much smaller factor of 1.17 rather than 3.01.

Most recently, Willi Geiselmann and Rainer Steinwandt described at PKC'03 [4] a hardware device for supporting the sieving step in integer factoring algorithms such as the quadratic sieve or the number field sieve. In analogy to Bernstein's proposal the device relies on a mesh of very simple processing units which can be manufactured on standard wafers with 200mm or 300mm diameter. The authors state that their device might outperform a TWINKLE device for factoring a 512-bit number. The algorithm used within the sieving device is based on Schimmler's sorting algorithm. The number of transistors required per processing unit is estimated with 2,500 transistors. With current $0.13\mu\text{m}$ technology, on the square area of a 300mm wafer, 7.2 million processing units can be placed. To sieve the complete region necessary for a 512-bit factorization with such a device, less than 4 days are required.

The recent attack proposals on RSA with dedicated hardware look encouraging from a cryptanalytical viewpoint. Even though non of the more promising hardware-driven factorization machines have been built (or even partially built), there is a chance that they could pose a threat to RSA with currently used bit length, especially for RSA with 1024 bits. It appears to be highly advisable to research the area of special purpose factorization hardware further, in order to obtain valid statements about the security of RSA with currently used bit length. Moreover, it should be investigated to which extend index calculus attacks can benefit from special purpose hardware. Given the close coupling of engineering and mathematical issues, this is a particularly interesting but also challenging research area.

References

- [1] D. J. Bernstein. Circuits for integer factorization: A proposal, 2001.
- [2] M. Bond and R. Clayton. Efficient Uses of FPGA for DES and its experimental linear cryptanalysis. In Çetin K. Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, 4th International Workshop, CHES 2002 Proceedings*, Lecture Notes in Computer Science. Springer, 2002.
- [3] EFF. Website of the electronic frontier foundation, 1998.
- [4] W. Geiselmann and R. Steinwandt. A Dedicated Sieving Hardware. In Y.G. Desmedt, editor, *Public Key Cryptography, 6th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2003 Proceedings*, volume 2567 of *Lecture Notes in Computer Science*, pages 254 – 266. Springer, 2002.
- [5] I. Hamer and P. Cho. DES Cracking on the Transmogrifier 2a. In Çetin K. Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, 1st International Workshop, CHES 1999 Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 13 – 24. Springer, August 1999.
- [6] H. J. Kim and W. H. Mangione-Smith. Factoring large numbers with programmable hardware. In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, pages 41–48. ACM Press, 2000.
- [7] A. K. Lenstra, A. Shamir, J. Tomlinson, and E. Tromer. Analysis of bernstein’s factorization circuit. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*. Springer, 2002.
- [8] A. Shamir. Factoring large numbers with the TWINKLE device. In Çetin K. Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, 1st International Workshop, CHES 1999 Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 2 – 12. Springer, August 1999.
- [9] M. J. Wiener. Efficient DES key search. Technical Report TR-244, Carleton University, 1996.

CrOPTOgraphy. CrOPTOgraphy [4] is the generic name that has been given to attack techniques that use optical devices instead of binary computers.

The first such device that has been designed to break a cryptographic scheme is Shamir’s Twinkle device [3]. This device is supposed to help to factor large numbers. It is a sieving device that can replace the most time-consuming step of the best-known factoring algorithm, NFS. The basic idea of this device is that a large amount of LED is being controlled, each by a small program such that it is lit to indicate that the current common value is a multiple of some fixed prime number. The intensity of the LED corresponds to that prime number and the device detects a smooth number if the total intensity of the LED is above some threshold.

The key advantage of this device is that the addition of the intensities is made in one cycle, which is not possible with a binary computer.

Another optical device has first been designed to make efficient secret sharing [1]. Each user has a transparency with black dots and transparent holes. By stacking many of these transparencies, one can easily detect which points are simultaneously transparent for all these transparencies. This device can be used for secret sharing and its key advantage is that the logical OR of many black dots is made instantaneously.

Another application of this technique is the visual cryptanalysis [2] where massively parallel computations are made with such transparencies. A photographic film can translate the result of a parallel OR to a unique transparency, and the logical NOT can be implemented with negative film.

While none of these cryptanalytic techniques apparently has been used in practice, the massive parallelism that optical devices can achieve at a low cost can be a threat to some cryptographic techniques.

References

- [1] M. Naor and A. Shamir. Visual Cryptography. Eurocrypt'94, pp. 1-12, LNCS 950, Springer, 1995.
- [2] A. Shamir. Visual Cryptanalysis. Eurocrypt'98, pp. 201-210, LNCS 1403, Springer, 1998.
- [3] A. K. Lenstra and A. Shamir. Analysis and optimization of the TWINKLE factoring device. Eurocrypt'00, pp. 35-52, LNCS 1807, Springer, 2000.
- [4] A. Shamir. New Directions in Cryptography. Invited talk at CHES'01.

5 Mathematical foundations

5.1 Theory of computation

5.1.1 Complexity theory

Complexity theory studies the amount of resources (computation time, memory, etc), that is necessary and/or sufficient to solve computational problems. From cryptographic point of view, it tries to give an answer to the fundamental question: “Is cryptography at least theoretically possible?” Intuitively, cryptography requires one-way functions: functions easy to compute, but hard to invert. However, whereas “hardness” in complexity theory usually means worst-case hardness, for cryptography one requires at least some amount of average-case hardness, [11]. Thus, existence of one-way functions would imply $P \neq NP$ and resolve the perhaps deepest of all problems in the theory of computing. Not surprisingly, we are therefore currently very far from showing the existence of one-way functions. Some initially promising work on basing cryptography on NP-hard problems, has later turned out to have limited implications, [14]. It is trivial to see that almost all functions require exponential

size circuits. Yet, the best known lower bound for an explicit function is linear, and the currently best complexity relation between an explicit function and its inverse only guarantees a constant factor, [12]. Relative to certain computational models, better lower bounds are known for some explicit functions. For instance, lower bounds in terms of linear and algebraic complexity for e.g. discrete logs (or parts thereof) have been established, see e.g. [10].

On the other hand, if one starts by assuming that one-way functions do exist, impressive progress has been made over the last twenty years. Fundamental results being e.g. [16, 3, 15, 9] to mention a few. We now know that one-way functions (or some strengthened form thereof) is not only necessary, but also sufficient for most cryptographic problems. Probably for this reason, it seems that research aiming towards establishing cryptographic primitives based on general one-way functions has now more or less come to a stand-still. Of course, as mentioned many of the problems have indeed been solved but there are still some important open problems. There is also room for improving the existing ones in many cases, in particular in terms of complexity of construction. These may be needed as factoring and discrete logs may eventually turn out not to be as hard as we think, and we then need to resort to more general constructions.

Assuming that we have a one-way function $f(x)$, which per se only guarantees some amount of difficulty of *complete* inversion, the study of whether approximations of parts of x can be found has drawn a lot of attention. In [5] it was shown that any one-way function must have at least $O(\log n)$ randomized predicates that are hard. Explicit functions such as RSA, discrete logs, DH etc, have also been studied [4, 8] with regard to such properties, and there, non-randomized hard predicates are known.

For two of the main assumptions in public-key cryptography, namely the Diffie-Hellman assumption and the discrete logarithm problem, [13] provide a reduction which shows polynomial time equivalence under a certain plausible assumption.

In the study of cryptographic protocols and multi-party computation, while some highly relevant and good work in establishing sound models and secure protocols has seen light of day, e.g. [7], we have unfortunately also seen a trend of developing rather ad-hoc solutions to specific problems where in addition lots of the existing proofs are quite incomplete or adopt models and assumptions that in some cases turned out to be unrealistic. As many of the protocols are aimed at solving everyday problems (voting, payments,.. etc) it would be very dangerous to adopt these without fully understanding the security.

Zero-knowledge techniques, introduced in [6], have evolved during the last two decades. These have indeed turned out to have many practical applications, e.g. in identification protocols.

A very positive trend is turning theoretical solutions into practical ones by adopting the basic theories developed, making them concrete. For instance proving the concrete security of block-cipher modes, modeling ciphers by pseudo-random permutations developed in complexity theoretic cryptography, is one such result, see [1]. This joining of theory and practice has been very fruitful. However, there also seems to be confusion on what the theoretical results say when trying to use them in practice. For example, concerning the bit-security of the RSA function, and the implied pseudo-random number generators derived by iterating $x_{j+1} = \text{RSA}(x_j)$, one of the most popular and most cited cryptography books gives reference to a number of research papers, and from that concludes that “the least significant $\log_2 n$

bits of x_j can be used [as pseudo-random output from each x_j]. At best, such a statement is only unjustified, at worst it tricks practitioners into making insecure protocols. The exact statement, as claimed above, has never been established; the result referred to (i.e. [2]) is first of all asymptotic, and secondly, there is an implied constant in front of “ $\log_2 n$ ” that, to our knowledge, has never been explicitly evaluated. Though such an RSA generator might be very attractive in practice, we believe that it is very dangerous to draw such absolute conclusions from complexity theoretical results about the hardness of RSA.

References

- [1] M. Bellare, A. Desai, E. Jorjani, and P. Rogaway: “A Concrete Security Treatment of Symmetric Encryption”, In proceeding of 38th IEEE FOCS, pp. 394–403, 1997.
- [2] L. Blum, M. Blum, and M. Shub: “A Simple Secure Pseudo-Random Number Generator”, SIAM J. of Comput. 15(2):364–383, 1986.
- [3] M. Blum and S. Micali: “How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits”, SIAM J. Comput. 13(4):850–864, 1984.
- [4] D. Boneh and R. Venkatesan: “Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and related schemes”, In Proceedings Crypto ’96, LNCS 1109, pp. 129–142, 1996.
- [5] O. Goldreich and L. Levin: “A Hard-core Predicate for all One-way Functions”, In Proceedings of 21st ACM STOC, pp. 25–32, 1989.
- [6] S. Goldwasser, S. Micali, and C. Rackoff, “The Knowledge Complexity of Interactive Proof-systems”, SIAM J. Comput. 18(1):186–208, 1989.
- [7] O. Goldreich, S. Micali, and A. Wigderson: “How to Play any Mental Game”, In Proceedings of 19th ACM STOC, pp. 218–229, 1987.
- [8] J. Håstad and M. Näslund, “The Security of all RSA and Discrete Log Bits”, J. ACM (to appear). Prel. version in proceedings of 39th IEEE FOCS, pp. 510–519, 1998.
- [9] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby: “Construction of a Pseudo-random Generator from any One-way Function”, SIAM J. Comput. 28(4):1364–1396, 1999.
- [10] T. Lange, S. V. Konyagin, and I. E. Shparlinski: “Linear Complexity of the Discrete Logarithm”, Designs, Codes and Cryptography (to appear).
- [11] L. A. Levin: “Average Case Complete Problems”, SIAM J. Comput., 15(1):285–286, 1986.
- [12] J. Massey: “The Difficulty with Difficulty”, 1996 IACR Distinguished Lecture, <http://www.iacr.org/publications/dl/massey96/>
- [13] U. Maurer and S. Wolf: “The Relationship Between Breaking the Diffie-Hellman Protocol and Computing Discrete Logarithms”, SIAM J. Comput. 28(5):1689–1721, 1999.

- [14] P. Q. Nguyen and J. Stern: “Cryptanalysis of the Ajtai-Dwork Cryptosystem”, Proceedings, Crypto ’98, LNCS 1462, pp. 223–242, 1998.
- [15] J. Rompel: “One-way Functions are necessary and sufficient for Secure Signatures”, In Proceedings of 22nd ACM STOC, pp. 387–394, 1990.
- [16] A. C. Yao: “Theory and Applications of Trapdoor Functions”, In Proceedings of 23rd IEEE FOCS, pp. 80–91, 1982.

5.1.2 Information theory

Information theoretic cryptography has aimed to provide us with cryptographic primitives that are *unconditionally secure* against arbitrarily powerful adversaries. Often, this means that the adversary is guaranteed to have no advantage at all, but some work provides a (non-zero) upper bound on the advantage. In a sense, modern cryptography started as information theoretic cryptography with the seminal work of Shannon, first classified, but eventually published in the late 40’s, [21]. In the last twenty years, the area has been taken up again, and some very interesting results have been obtained. From a practical point of view, the area seems particularly interesting in applications aimed at protecting users’ privacy.

Basically, recent information theoretic cryptography has been concerned with protocol problems of two kinds: *unconditionally secure key-exchange and key-distribution*, and more general *multi-party computation* problems with privacy and robustness guarantees. Some work on provably secure ciphers (with limited practical value) and provably secure message authentication (with highly practical value) has also been done.

For the key-exchange problem, some interesting results have been obtained using authentic, noisy channels. These protocols typically use three phases: advantage distillation (Alice and Bob obtains an information advantage relative to Eve), information reconciliation (Alice and Bob agree on a shared string, S , using error correction), and privacy amplification (Alice and Bob compress S to a shorter string, but with “full” entropy relative to Eve). Early work was based on the assumption that Eve’s channel was noisier than the legitimate parties’ common channel. Later, it was shown that even if Eve has a superior channel quality, key-exchange is still possible as long as the errors on Eve’s channel are (to some extent) independent of those on the Alice and Bob’s channel. A related, generalized model that has been studied is based on correlated randomness together with public discussion. In this latter case, all parties have access to a common “random oracle”, providing bits over noisy channels to the the users (and adversaries). The legitimate users are in addition connected by an authentic but open channel. Interesting quantitative aspects here are *secrecy capacity* and *secret-key rate* of the protocols. These quantities basically tells us at which rate secret keys can be created, while keeping the eavesdropper’s advantage arbitrarily small. Upper and lower bounds on these quantities have been studied. We refer to [16] for a good overview and exact references.

Another two-party protocol that has gained quite a bit of study is the information theoretic variants of *zero-knowledge*, where protection of the secret witness is guaranteed (at least to some extent). Very general results based on information theoretic “blobs” for bit-commitment show that satisfiability of any boolean function can be proven with *minimum disclosure*, [5]. Various implementations of these blobs have been proposed, some based on quantum

physics, other (somewhat weaker, but perhaps more realistic) are based on number theory. The different blob constructions offer different types of protection for the verifier and prover, respectively.

For the general multi-party computation problem, there has basically been studies of two models, assuming either pairwise *secure and authentic channels*, or a *perfect information model*, where only broadcast communication is possible.

For the (Byzantine) secure channels model, results similar to those obtained in the computational model have been established, [1]. With passive adversaries, an honest majority (out of n parties) is necessary and sufficient to compute any function, whereas with active adversaries, an honest two-thirds majority is needed. Thus, these bounds are tight. There is also a zero-one law: if we call a given function t -private, if it can be securely compute in the presence of $\leq t$ “bad” parties, it is known that a function is either $\lfloor \frac{n-1}{2} \rfloor$ -private, or, it is (completely) n -private. Moreover, the only $\lceil \frac{n}{2} \rceil$ -private (hence also n -private) functions are functions of form $f(x_1, \dots, x_n) = f_1(x_1) \oplus \dots \oplus f_n(x_n)$, as one perhaps would suspect, see [11].

Some generalized models have been studied where one takes into account more complex forms of “bad behavior”. For instance, one can consider the case where t_a parties are actively bad, t_p are passively corrupt, and t_f parties might stop, refusing to co-operate. Also here, a tight bound is known, stating that secure multi-party computation is possible if, and only if, $3t_a + t_p + t_f < n$ and $2t_a + 2t_p + t_f < n$. In yet another model, one is analyzing the case when the adversary has a control structure: i.e. one considers collections of sub-groups of parties under the adversary’s control, rather than the total number of controlled parties. It is shown that if control is active (passive), secure multi-party computation is possible if no two (three) sets of the control structure cover the whole set of parties. All these results nicely characterize what is possible and what is not, see [12, 14].

In the perfect information model, as far as we understand, there has mainly been studies performed on the leader-election (and collective coin-flipping) problem. Here, some fraction of users constitute a “bad” coalition and one strives to minimize the number of rounds and communicated bits per user per round, while guaranteeing that the elected leader is honest with the “right” probability. For collective coin-flipping, the goal is to agree on an unbiased coin-flip (which thus can be reduced to leader election). In the one-bit per round case, it is for instance known that $\log^* n$ rounds are needed, [19]. While some progress has been made in recent years, there are gaps between upper and lower bounds.

The perfect information model is also related to some properties of boolean functions that have been studied independently. First the study of *influence of variables on boolean functions*: is there a fixed set of inputs of certain (small) size that completely determine the boolean output? An important result can be found in [4]. Generalizing this is the problem of *t -resilient functions*, where one considers functions $\{0, 1\}^n \rightarrow \{0, 1\}^m$. What is the maximum size, t , of any input set the adversary can control, so that still (as a function of the remaining inputs) the output is uniform in $\{0, 1\}^m$? Exact values and explicit constructions are known for $m = 1, 2$ and some “special” values of (n, m) , e.g. [9]. The general problem seems quite open, though some upper/lower bounds and explicit construction approximating the optimum exist.

Another specific protocol problem considered is the *private information retrieval* problem:

a user remotely stores n bits of information and wants to retrieve the i th bit, without disclosing what bit he wants with perfect privacy. Whereas $\Omega(n)$ bits of communication (the whole database) is needed with a single database server, with the database distributed over k servers, it is possible to solve the problem using $O(n^{1/(2k-1)})$ bits of communication, see [10].

One of the most well-known and widely studied problem in the area is (perfect) *secret sharing* (SS), [20, 2], and variants thereof such as threshold SS, verifiable SS, generalized (access structure based) SS, etc. A quantitative aspect is the *rate* (size of secret/size of share) of the scheme. Very simple perfect and ideal (having rate 1) schemes are known for standard threshold SS, but for generalized SS, examples of access structures are known where ideal schemes do not exist. *Visual cryptography* [18], and later generalizations thereof, is a clever idea to perform a more “human-readable” form of SS.

Related to SS is some more specific work on *key-distribution schemes*. One wishes to distribute as little keying material as possible, while allowing any pair out of n users to derive from it a key, secure against coalitions of $\leq j$ other users. Here, lower bounds and some optimal constructions based on MDS codes are known, [3].

While Shannon’s *one-time pad* (proposed already in 1926 by Vernam, but without security proof) gives perfect secrecy, it provides no authenticity/integrity whatsoever. However, also integrity can be perfectly achieved. Carter and Wegman’s results on *universal hash function families* (UHF), [6], provides us with information theoretically secure message authentication codes. More recent work has therefore mainly been aimed at optimizing constructions of UHF. For (strong) UHF, an essentially tight lower bound on their circuit complexity has been established. However, subsequent work has relaxed the condition on strong universality, thereby significantly improving parameter sizes and efficiency, e.g. using algebraic curves. Highly practical constructions of so-called ϵ -almost- Δ *universal families* are known, e.g. MMH, [13].

Related to authentication, we mention *blind signatures*, [7], where anyone can verify the validity, but the signer cannot tell to whom he issued a certain signature. Whereas the security of these signatures so far is only computational, the privacy of the user remains unconditional. These schemes have seen many applications, e.g. in voting and payment protocols.

One-time pad is not the only provably secure cipher. Rip van Winkle (RvW) [15] is another construction that guarantees some amount of unbreakability. This cipher is, however, completely impractical. A randomized stream-cipher [17], based on RvW and huge amounts of publicly available random bits has also been proposed and is “conceivably practical”. Here, an adversary is unable to obtain any information whatsoever about the plaintext with probability arbitrarily close to 1, unless the adversary can perform an infeasible computation.

In summary, information theoretic cryptography has provided us not only with interesting theoretic “artifacts”, but also highly desirable privacy primitives, as well as very efficient authentication codes. It could, perhaps, also turn out to be an alternative to quantum cryptography for some key-exchange applications.

References

- [1] M. Ben-Or, S. Goldwasser, and A. Wigderson: “Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed”, In Proceedings of the 20th ACM STOC, pp. 1–10, 1988.
- [2] G. R. Blakely: “Safeguarding cryptographic keys”, In Proceedings of AFIPS 79, pp. 313–317, 1979.
- [3] R. Blom: “An optimal class of symmetric key generation systems”, In Proceedings of EUROCRYPT 84, LNCS 209, pp. 335–338, 1985.
- [4] J. Bourgain, J. Kahn, G. Kalai, Y. Katznelson, and N. Linial: “The influence of variables in product spaces”, *Israel Jour. Math.* 77:55–64, 1992.
- [5] G. Brassard, D. Chaum, and C. Crepeau: “Minimum disclosure proofs of knowledge”, *JCSS* 37(2):156–189, 1988.
- [6] I. L. Carter and M. N. Wegman: “Universal classes of hash functions”, *JCSS* 18(2):143–154, 1979.
- [7] D. Chaum: “Blind signatures for untraceable payments”, In Proceedings of Crypto '82, pp. 199–203, 1983.
- [8] D. Chaum, C. Crepeau, and I. Damgard: “Multiparty Unconditionally Secure Protocols”, In Proceedings of the 20th ACM STOC, pp. 11–19, 1988.
- [9] B. Chor, O. Goldreich, J. Hastad, J. Freidmann. S. Rudich, and R. Smolensky: “The Bit Extraction Problem or t-Resilient Functions”, In Proceedings of 26th IEEE FOCS, pp. 396–407, 1985.
- [10] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan: “Private Information Retrieval”, *J. ACM* 45(6):965–981, 1998.
- [11] B. Chor and E. Kushilevitz: “A Zero-One Law for Boolean Privacy”, *SIAM J. on Disc. Math.* 4(1):36–47, 1991.
- [12] M. Fitzi, M. Hirt, and Ueli Maurer, “General Adversaries in Unconditional Multi-Party Computation”, In Proceedings of ASIACRYPT '99, LNCS 1716, pp. 232–246, 1999.
- [13] S. Halevi and H. Krawczyk: “MMH: Software Message Authentication in the Gbit/second Rates”, Proceedings of the 4th FSE, LNCS 1267, pp. 172–189, 1997.
- [14] M. Hirt and U. Maurer, “Complete Characterization of Adversaries Tolerable in Secure Multi-Party Computation”, In Proceedings of 16th PODC, pp. 25–34, 1997.
- [15] J. L. Massey and I. Ingemarsson: “The Rip van Winkle cipher: A simple and provably computationally secure cipher with a finite key”, *Proc. 1985 IEEE ISIT*, pp. 146, 1985.
- [16] U. Maurer: “Information-Theoretic Cryptography”, In Proceedings of CRYPTO '99, LNCS 1666, pp. 47–64, 1999.

- [17] U. Maurer: “Conditionally-Perfect Secrecy and a Provably-Secure Randomized Cipher”, J. Cryptology, 5(1):53–66, 1992,
- [18] M. Naor and A. Shamir: “Visual Cryptography”, In Proceedings of Eurocrypt 94, LNCS 950, pp. 1–12, 1995.
- [19] A. Russell, and D. Zuckerman: “Perfect Information Leader Election in $\log^* n + O(1)$ Rounds”, JCSS 63(4):612–626, 2001.
- [20] A. Shamir: “How to Share a Secret”, CACM 22(11):612–613, 1979.
- [21] C. E. Shannon: “Communication theory of secrecy systems”, Bell System Technical Journal 28:656–715, 1949.

5.2 Combinatorics in Finite Fields

Finite fields appear in many branches of cryptography. Elliptic curves over finite fields and subgroups of the multiplicative group of finite fields are used for digital signatures and key exchange. Finite fields are the base structure for combinatorial constructions which are applied in key sharing schemes. Another kind of example are S-boxes of block ciphers like for instance Rijndael and Misty, which are based on power functions in a field of characteristic 2. After all, the ring of integers modulo the product of two primes used in the RSA scheme is isomorphic to the direct product of two prime fields.

It is not over-emphasized to say that finite fields form the most important algebraic structure as a tool for cryptographic constructions. Therefore the study of finite fields, often in connection with combinatorial properties, was and will remain a central issue of research in cryptography.

The past years have seen new research development in this area mainly in relation to symmetric techniques, namely in the construction of S-boxes for block ciphers. Properties of Boolean (vector) functions (mappings from $GF(2)^n$ to $GF(2)^m$) like high non-linearity have been studied. This property is necessary to withstand linearization attacks where the adversary tries to model the substitution box by a linear function. Presently we have a list (see [2]) of exponents d such that x^d is almost perfect nonlinear (APN) for a family of fields. For extension degrees ≤ 25 this list is known to be complete. The proofs to show the APN property usually require to show that a related polynomial is a permutation polynomial. The general description of a powerful technique which has been used in the latest proofs can be found in [1].

References

- [1] H. Dobbertin, *Uniformly representable permutation polynomials*, Sequences and their Applications, Proceedings of SETA 01 (Editor: T. Hellesteth, V. Kumar, K. Yang), Springer-Verlag.
- [2] H. Dobbertin, *Almost perfectly nonlinear power functions on $GF(2^n)$: a new case for n divisible by 5*, Proceedings of the Fifth Conference on Finite Fields and Applications Fq5, (Editor: D. Jungnickel und H. Niederreiter), Springer-Verlag, 2001, S. 113 -121.

5.3 Algorithmic number theory

5.3.1 Integer factorization

Integer factorization is arguably the most famous problem in algorithmic number theory. Many public-key cryptosystems, including RSA, require integer factorization to be hard. The number field sieve (NFS) is still the fastest factoring algorithm known: the current factoring record is the factorization of a 158-digit cofactor of $2^{953} + 1$, established in 2002 by Bahr, Franke and Kleinjung using GNFS. In the past ten years, research has tried to improve the number field sieve, but no new major factoring algorithm has been discovered. The NFS can be divided in several parts, which have all been improved since the appearance of the NFS:

- Polynomial selection. Finding “good” polynomials. At the beginning, polynomials were chosen to simply minimize the size of their coefficients. Now, polynomials are also chosen to have good root properties: they should have many roots modulo many small primes.
- Sieving. This is the core of the algorithm, where one tries to find smooth polynomial values. There are currently two methods in use: line sieving and lattice sieving.
- Linear algebra. Sieving yields a huge system of linear equations over the binary field. In this stage, one would like to find non-trivial solutions to this system. Recent factorization records used a block Lanczos algorithm published by Montgomery at Eurocrypt ’95.
- Square root. In a final stage, one needs to extract square roots of huge algebraic numbers. In recent records, one used a very efficient heuristic algorithm developed by Montgomery.

Very recently, some non negligible improvements to the number field sieve have been found: Franke and Kleinjung [5] improved both the polynomial selection method of Montgomery and Murphy, and Pollard’s lattice sieving, which should lead to new factorization records. In parallel, there has been a lot of discussion about a new cost model introduced by Bernstein last year to analyze the complexity of the number field sieve.

There has also been a lot of discussion about the very recent result of Agrawal, Kayal, and Saxena [1] who solved in August 2002 a long-standing conjecture in algorithmic number theory: primality is in P; that is, one can decide if a given number is prime in time polynomial in the bit-length of the number. However, this is almost irrelevant to cryptography. From a security point of view, the primality problem is completely independent from the factorization problem. And from a practical point of view, the result of Agrawal, Kayal and Saxena is useless for now: the primality problem is not considered a practical issue in cryptography, because one already knows very efficient probabilistic algorithms that can decide (with negligible error probability) if a given number is prime or not.

5.3.2 Discrete logarithm in finite fields

The hardness of the discrete logarithm problem can be used to build public-key cryptosystems, such as El Gamal. The fastest algorithms known to extract discrete logarithms in finite fields

are the function field sieve (in small characteristic) and the number field sieve (in small degree), which are both analogues of the number field sieve factoring algorithm. In the past ten years, there has been two kinds of research on the discrete logarithm in finite fields:

- Improving algorithms to solve the discrete logarithm problem. Although no new major discrete log algorithm has been discovered, there has been a few practical improvements to the function field sieve and the number field sieve, which have led to the current records.
- Shortening key sizes, by finding compact representations in certain finite fields. This was first done in the LUC cryptosystem, then in XTR and GH. Such schemes make use of certain subgroups of \mathbf{F}_{p^r} , $r = 2, 3, 6$ which allow faster arithmetic than the field \mathbf{F}_{p^r} itself. So far the security of these systems seems to be the same as of generic DL systems with the same parameters.

5.3.3 Algebraic curves over finite fields

Throughout the past years there has been quite some work done on cryptography based on curves over finite fields, because the Jacobian of an algebraic curve is an interesting group from a cryptographic point of view: in particular, the discrete logarithm problem may be much more difficult than in the multiplicative group of a finite field, which would lead to much smaller parameters and a better efficiency. In the “simple” case of elliptic curves, the Jacobian can be identified with the set of points of the curve. Research has focused on the following topics:

Point counting: In general, to use algebraic curves in cryptography, one first needs to know the cardinality of the Jacobian. There are basically two ways to tackle this issue: one is to select special curves for which point counting is easier than with random curves, the other is to use a general point counting algorithm. The first approach may lead to security troubles. Hence, there should be good reasons to use such curves e.g. faster arithmetic and a good estimate on the possible weaknesses. The second approach is more and more popular, thanks to dramatic improvements in point counting techniques in the elliptic curve case (genus one): there is now ongoing research to extend such techniques to arbitrary curves with genus ≥ 1 . In 1985, Schoof found the first polynomial-time algorithm to count points on elliptic curves over finite fields of large characteristic, which was later extended to arbitrary genus curves by Pila. However, although the algorithm was efficient from a theoretical point of view, it was far from being practical: Elkies and Atkin later found crucial improvements to make it practical, and the resulting algorithm is now known as the Schoof-Elkies-Atkin algorithm (SEA). SEA was adapted to the small characteristic case by Couveignes. In the past few years, there has been dramatic improvements in the small characteristic case, using a different approach based on p -adic arithmetic: new algorithms, more efficient and much simpler to implement, have been found, notably by Kedlaya, Mestre and Satoh. Some of them can be extended to hyperelliptic curves of larger genus (still in small characteristic). There is ongoing research to optimize such algorithms. A way to avoid point counting is to construct the curve via complex multiplication. Here, one starts with the group order and then

constructs the curve. This method was suggested for elliptic curves by Atkin and is now available also for larger genus curves.

Discrete logarithm: The main interest of algebraic curves in cryptography comes from the fact that when the curve is carefully chosen, the best attack known to solve discrete logs is Pollard's rho method or a variant of it, whose running time is exponential in the size of the ground field. However, a few classes of weak curves have been found, in the sense that the discrete logarithm problem in such curves is either easy, or no more difficult than in a finite field of comparable size, in which case there is no advantage to use such curves. These classes are supersingular curves (or more generally, all curves weak under the Frey-Rück attack), curves over prime fields where the prime divides the group order (Rück attack) and due to the paper by Gaudry, Hess, and Smart also some curves over extension fields of composed degree which can be attacked by Weil descent. In the general case, Adleman, DeMarras and Huang showed in 1994 that when the genus is sufficiently large compared to the size of the ground field, one can compute discrete logs in subexponential time. Gaudry [3] found in 2000 an algorithm more efficient than Pollard's rho method against curves of genus ≥ 5 . Thus, only curves of genus ≤ 4 have potential cryptographic interest, at least if one is interested in reducing keysize.

Pairings: One important new trend is the positive use of pairings (such as those of Weil and Tate) in cryptography. Such mathematical objects were first used in cryptography in a destructive sense, to show that the discrete logarithm problem over supersingular elliptic curves was not harder than the discrete logarithm problem over finite fields. The first positive applications were found independently by Joux and Kasahara, Mitsunari, Oghishi and Sakai. Very interesting applications have been found since, including an identity-based public-key cryptosystem by Boneh and Franklin [4], and a short signature scheme by Boneh, Lynn and Schacham. There is now ongoing research to improve implementation aspects, to find new cryptographic applications, to find more curves for which the pairing can be used, and to generalize pairings to multilinear forms (which would have several applications).

Larger genus curves: As already mentioned in the discrete log part, we may restrict to curves of genus ≤ 4 . The genus one case, *i.e.* elliptic curves, is well-understood, in the sense, that we know how to perform the group operation very efficiently, and we know how to count points efficiently. But difficulties arise in the general case, and there is ongoing research to tackle such issues. For hyperelliptic curves we now have explicit formulae to perform group operations with similar efficiency as for elliptic curves and we also have alternative systems of coordinates for genus 2 curves. However, this is still missing for more general curves and we still have space for improvements in the hyperelliptic curve case. Furthermore, real quadratic function fields can lead to fast systems. To speed up the arithmetic on curves, endomorphisms can be exploited, such as in the so-called Koblitz curves: this has been studied for elliptic and arbitrary genus curves.

5.3.4 Geometry of numbers

Geometry of numbers is a branch of number theory dealing with lattices, which are regular arrangements of points in n -dimensional space. The goal of lattice basis reduction is to find interesting and useful representations of lattices. Until recently, lattice basis reduction was used to break various public-key cryptosystems, including knapsack cryptosystems and special cases of RSA. But the potential hardness of several lattice problems is now used as a security assumption for several public-key cryptosystems. Some of these cryptosystems offer a few computational advantages, compared to other cryptosystems known. In the past few years, research has focused on the following topics (see the survey [6]):

- Complexity aspects of lattice problems. Two breakthrough results by Ajtai in 1996 and 1997 has inspired a series of complexity results. More precisely, Ajtai discovered a worst-case/average-case equivalence for certain lattice problems, and he showed the NP-hardness under randomized reductions of the most famous lattice problem, the so-called shortest vector problem. There is ongoing research to fill the gap between hardness results and algorithmic results: we know that for several lattice problems, finding the exact solution of a very good approximation is hard, but we also know how to efficiently find certain approximations which are not too bad in theory, and which often turn out to be very good in practice.
- Using lattice problems to build cryptographic primitives. This was inspired by both the invention of the NTRU public-key encryption scheme by Hoffstein, Pipher and Silverman in 1996, and the Ajtai-Dwork cryptosystem in 1997 (which followed Ajtai's breakthrough results in complexity theory). Goldreich, Goldwasser and Halevi found a lattice analogue of the McEliece cryptosystem based on coding theory. However, recent cryptanalytic results (notably by Nguyen and Stern) have shown that, for now, among all lattice-based cryptosystems, only NTRU might be an alternative to more established cryptosystems such as RSA. One needs to be very cautious when comparing those schemes with more established schemes, because lattice-based schemes are still very recent. There is ongoing research to further our understanding of the security of NTRU, and to find other practical lattice-based cryptosystems.
- New algorithms to solve lattice problems. Since the breakthrough invention of the Lenstra-Lenstra-Lovász algorithm (LLL) twenty years ago, very few algorithms have been discovered. From a practical point of view, the most notable improvement is Schnorr's tradeoff algorithm between LLL reduction and Hermite-Korkine-Zolotarev reduction. But the field may be evolving: In 2001, Ajtai, Kumar and Sivakumar [2] found a new and more efficient (from a theoretical point of view) algorithm to solve several lattice problems, while Semaev has revisited the low-dimensional case. A major problem is to find new principles to design new algorithms.

5.3.5 Systems of multivariate polynomial equations

There has been several attempts to change the mathematics of RSA. In the eighties, Matsumoto and Imai proposed to consider systems of multivariate polynomial equations over a small finite field, instead of a univariate polynomial equation modulo a large number hard

to factor, like in RSA. The size of the finite field makes operations faster, but the size of the polynomial system enlarges the keysize. At Crypto '95, Patarin broke the Matsumoto-Imai cryptosystem, but he noticed that his attack could be prevented in several ways by slight variations on the basic scheme. This gave rise to a family of new public-key cryptosystems, usually called the HFE family for *hidden field equations*. With HFE cryptosystems it is possible to build signature schemes with low computing power (such as Sflash) and very short signatures (such as Quartz), which make them very suitable for smartcard applications. There are however two disadvantages: the public key is somewhat long, and the security is very difficult to study due to many security parameters. In the past few years, many people have studied the security of HFE schemes, and it has turned out that several variants of HFE can be broken in very different ways. There is ongoing research to further our understanding of the security of HFE schemes, and to build new cryptographic primitives out of multivariate polynomials.

References

- [1] M. Agrawal, N. Kayal, N. and N. Saxena, PRIMES is in P. Preprint, date 6.8.2002.
- [2] M. Ajtai, R. Kumar and D. Sivakumar, 'A sieve algorithm for the shortest lattice vector problem' *Proc. 33rd ACM Symp. on Theory of Comput.*, ACM, 2001, 601–610.
- [3] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. *Lect. Notes Comput. Sci.* 1807, Springer-Verlag (2000), pp. 19–34.
- [4] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. *Advances in cryptology – Crypto '01*, *Lect. Notes Comput. Sci.* 2139, Springer-Verlag (2001), pp. 213–229.
- [5] J. Franke and T. Kleinjung. Recent progress in GNFS factorization. Talk at ECC '02, September, 2002.
- [6] P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Cryptography and Lattices – Proc. CALC '01*, volume 2146 of *Lecture Notes in Computer Science*, pages 146–180. Springer-Verlag, 2001.

5.4 Combinatorial Group Theory

Due to Shor's algorithms for computing prime factorizations and discrete logarithms on quantum computers [8], most of nowadays used public key cryptosystems had to be considered as insecure, if sufficiently large quantum computers became available. Currently, it is not possible to apply Shor's algorithms to cryptographically relevant parameters, as no suitable implementations of quantum computers are available yet. However, when dealing with long-term security, then quantum computers should be taken into account, and thus the question for mathematical primitives arises, that can serve as a basis for public key cryptography on 'classic' computers and where no structural attack based on the use of quantum computers is known.

One interesting line of research in this direction is the use of computational problems in non-abelian groups. Several interesting proposals for public key primitives have been put forward here: e. g., in [7] a possible generalization of the well-established ElGamal encryption scheme using non-abelian groups is considered; also an approach for constructing trapdoor-permutations by means of so called *logarithmic signatures* for finite non-abelian groups is proposed there. Other examples are provided by the work in [5, 1] where public key encryption schemes and key agreement protocols based on braid groups are considered. The latter groups are infinite, but allow for a finite presentation and can be handled conveniently in software implementations (cf. [3]). However, recent cryptanalytic results (e. g., [2, 4, 6]) show that more research is still needed in this area, before it is clear whether and to what extent reliable and practical cryptographic schemes can be derived from such group-theoretical problems.

References

- [1] Iris Anshel, Michael Anshel, Benji Fisher, and Dorian Goldfeld. New Key Agreement Protocols in Braid Group Cryptography. In David Naccache, editor, *"Topics in Cryptology — CT-RSA 2001"*, volume 2020 of *Lecture Notes in Computer Science*, pages 13–27. Springer, 2001.
- [2] Jens-Matthias Bohli, María Isabel González Vasco, Consuelo Martínez, and Rainer Steinwandt. Weak Keys in MST_1 . Cryptology ePrint Archive: Report 2002/070, 2002. <http://eprint.iacr.org/2002/070/>.
- [3] Jae Choon Cha, Ki Hyoung Ko, Sang Jin Lee, Jae Woo Han, and Jung Hee Cheon. An Efficient Implementation of Braid Groups. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 144–156. Springer, 2001.
- [4] Dennis Hofheinz and Rainer Steinwandt. A Practical Attack on Some Braid Group Based Cryptographic Primitives. In Yvo G. Desmedt, editor, *6th International Workshop on Practice and Theory in Public Key Cryptography, PKC 2003 Proceedings*, volume 2567 of *Lecture Notes in Computer Science*, pages 187–198. Springer, 2003.
- [5] Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju sung Kang, and Choonsik Park. New Public-Key Cryptosystem Using Braid Groups. In Mihir Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 166–183. Springer, 2000.
- [6] Eonkyung Lee and Je Hong Park. Cryptanalysis of the Public-key Encryption Based on Braid Groups. In *Proceedings of EUROCRYPT 2003*, Lecture Notes in Computer Science. Springer, 2003. To appear; see also http://crypt.kaist.ac.kr/pre_papers/LeePark.pdf.
- [7] Spyros S. Magliveras, Douglas R. Stinson, and Tran van Trung. New Approaches to Designing Public Key Cryptosystems Using One-Way Functions and Trapdoors in Finite Groups. *Journal of Cryptology*, 15(4):285–297, 2002.
- [8] Peter Shor. Polynomial time algorithms for prime factorization and discrete logarithms on quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.