Cryptography 2000 ± 10 *

Ueli Maurer

Department of Computer Science
Swiss Federal Institute of Technology (ETH), Zurich
CH-8092 Zurich, Switzerland,
maurer@inf.ethz.ch

Abstract. Cryptography is both a fascinating mathematical science and a key technology for the emerging information society, with theory and applications being closely related. This article reviews some highlights of the past ten years of research in cryptography and its applications and discusses some of the main challenges for future research.
A major future research theme is to weaken the assumptions on which security proofs are based, in particular computational intractability assumptions, trust assumptions, and physical assumptions. These assumptions must be made explicit. Computation and communication are physical processes rather than mathematically idealized objects, involving for example quantum phenomena. This fact could have a profound impact on cryptographic research and implementations.

1 Introduction

1.1 Scope of this Article

The task set to the authors of articles in this volume was to review the past ten years of research in their respective fields, and to speculate about possible developments in the next ten years. Due to space limitations and the mathematical nature of cryptography, it is impossible to include sufficient material for the paper to be a self-contained tutorial, nor a comprehensive overview of all important results of the past decade; the selection of topics is bound to be incomplete. The treatment is informal rather than technical, and the main focus is on identifying important issues and trends. We refer the newcomer to the books [65, 75, 85] and to Rivest's 1990 tutorial [73], which can also serve as a good point of reference for this article to address new developments in the decade since then.

1.2 Information Security

The emerging global information infrastructure has a dramatic impact on the economy and the society at large. One of the main paradigm shifts is that information is becoming a crucial if not the most important resource. Information differs radically from other resources; for instance, it can be copied without cost, it can be erased without leaving traces, and its value can change very rapidly depending on time and context. Protecting the new resource information is a major issue in the information economy, in particular when considering the increasing exposure of information to all sorts of risks and attacks, including illegitimate access to information (e.g. network sniffing), modification or insertion of information (e.g. IP spoofing), denial of service attacks, traffic analysis, replay attacks, malicious software (e.g. viruses and Trojan horses), and even "social engineering".

* A preliminary version of this paper appeared in volume 2000 of Springer Verlag's Lecture Notes in Computer Science.

Three basic security requirements are (1) hiding information, e.g. the content of a message (confidentiality), the identity of a user (anonymity), or the existence of information (steganography); (2) authentication of information; and (3) controlling access to information. More complex security requirements include the ability to prove facts related to information, for instance knowledge of a piece of information without revealing it (e.g. a password), or agreement to a piece of information (e.g. a digital contract). New applications like voting over the Internet, digital payment systems, the protection of personal privacy, the software piracy problem, distributed databases, or the protection of digital intellectual property lead to new and more complex security requirements. Many of them still remain to be identified.

One of the fundamental problems in information security is the distinction between "good" and "bad" information, for instance between good or malicious software or e-mail attachments, or between good or bad network traffic (e.g. through a firewall). This distinction problem (e.g. deciding whether a piece of software meets a given specification) is undecidable and therefore can generally not have a clean solution.

In contrast, cryptography[1] is generally used to solve well-defined problems with a clean solution (except for the fact that the security can usually not yet be rigorously proven). The power of cryptography is generally unleashed only when a problem is well-defined. Therefore, formalizing information security problems and determining whether a cryptographic solution exists is an important part of cryptographic research. In fact, some of the most important, and perhaps the most fascinating contributions in cryptographic research are intriguing and sometimes paradoxical solutions to certain information security problems for which, at first sight, no solution seems to exist. These outstanding contributions deserve mentioning in the first place of such an article (see Section 2).

1.3 History: From an Old Art to a Young Science

Cryptography, and even more so cryptanalysis, has played an important role in history, for instance in both world wars. We refer to [47, 84] for very good accounts of the history of cryptography. Until after the second world war, cryptography can be seen as an art more than a science, mainly used for military applications, and concerned almost exclusively with encryption. The encryption schemes were quite ad-hoc with essentially no theory supporting their security. In sharp contrast, modern cryptography is a science with a large variety of applications other than encryption, often implemented by sophisticated cryptographic protocols designed by mathematicians and computer scientists, and used primarily in commercial applications and critical components of the information infrastructure.

There are perhaps two single most important papers which triggered the transition of cryptography from an art to a science. First is Shannon's 1949 paper "Communication theory of secrecy systems" [80], a companion paper of [79] where the foundations of information theory are laid. Second, and even more influential, is Diffie and Hellman's 1976 paper "New directions in cryptography" [27], in which they describe their discovery of public-key cryptography.[2]

[1] The term cryptography is often used in a narrower sense, namely for the design of cryptographic systems, in contrast to cryptanalysis, the term used for breaking systems; cryptology is often used as the term that covers cryptography and cryptanalysis.

[2] Ralph Merkle had previously proposed ideas that cover the general concept of public-key cryptography, but his ideas, less cleanly formulated and hardly practical, went unnoticed for a while [67]. In the late 90's, the British government announced that public-key cryptography was originally invented at the Government Communications Headquarters (GCHQ) in Cheltenham in the early 70's (see [84]) by James Ellis and Clifford Cocks, including what became known later as the Diffie-Hellman protocol and the RSA system.

1.4 The Role of Assumptions and Trust

The security of every cryptographic system depends on certain assumptions. An obvious assumption, which is usually not stated explicitly, is that randomness exists, i.e., that the world is not deterministic. Note that quantum theory was the first major physical theory in which randomness is an inherent feature. Moreover, one implicitly assumes that random values can be independent. This implies, for instance, that telepathy does not exist, at least that physically protected information (secret keys) cannot be accessed without the owner's consent.

There are cryptographic systems[3] that no amount of computation can break. Such a system is called *information-theoretically secure.* However, most systems could theoretically be broken by a sufficient amount of computation, for instance by an exhaustive key search. The security of these systems relies on the computational infeasibility of breaking it, and such a system is referred to as *computationally secure.* Since no general and useful proofs of the computational difficulty of any problem (let alone a cryptographically significant one) are known (see Section 3), computational cryptography relies entirely on computational intractability assumptions.

Some security proofs are based on physical assumptions, for instance that quantum theory is correct, that certain devices are tamper-resistant, or that a communication channel is subject to a minimal noise level (see Section 9).

Other crucial assumptions made explicitly in cryptography are about the trustworthiness of certain entities and/or system components.

The goal of cautious cryptographic design is to make these assumptions both explicit and as weak as possible. Reducing the necessary assumptions and trust requirements for achieving a certain security objective is a major theme of current and future research in cryptography.

2 Cryptography – Science of Paradoxes

Many cryptographic inventions are quite paradoxical. The first and best example of a paradox, mentioned in Section 1.2, is the discovery of public-key cryptography by Diffie and Hellman [27] who showed that two entities Alice and Bob, not sharing a secret key initially, can generate such a key by communicating solely over an authenticated but otherwise insecure channel.[4]

The protocol works as follows. Let G be a finite cyclic group with order $|G|$ generated by the element g, such that computing discrete logarithms in G to the base g (i.e., for a given $b \in G$, to compute $a \in \mathbf{Z}$ such that $g^a = b$) is computationally infeasible.[5] Alice and Bob secretly choose integers x_A and x_B, respectively, at random from the interval $[0, |G| - 1]$. Then they compute $y_A = g^{x_A}$ and $y_B = g^{x_B}$, respectively, and exchange these group elements over the insecure public channel. Finally, Alice and Bob compute $s_{AB} = y_B^{x_A} = g^{x_A x_B}$ and $s_{BA} = y_A^{x_B} = g^{x_B x_A}$, respectively. Since $s_{AB} = s_{BA}$, this quantity can be used as a secret key shared by Alice and Bob. A mechanical analog of the Diffie-Hellman key agreement protocol is shown in Figure 1.

Some other examples of paradoxical concepts in cryptography are described below. Diffie and Hellman in [27] also proposed the concepts of a *public-key cryptosystem* and of a *digital*

[3] e.g. the one-time pad that encrypts the binary plaintext sequence by adding (bitwise modulo 2) a completely random binary key sequence.

[4] In fact, the most important feature of public-key cryptography is that it allows to exploit channels that are only authenticated but not confidential. Such channels were useless for secrecy applications before public-key cryptography was invented.

[5] The discrete logarithms problem is today the most important cryptographic problem (e.g. see [76]). Whether it is feasible or infeasible depends on the group. The group operation proposed in [27] is multiplication modulo a large prime p; the corresponding group $\mathbf{Z}_p^*$ has $p - 1$ elements.

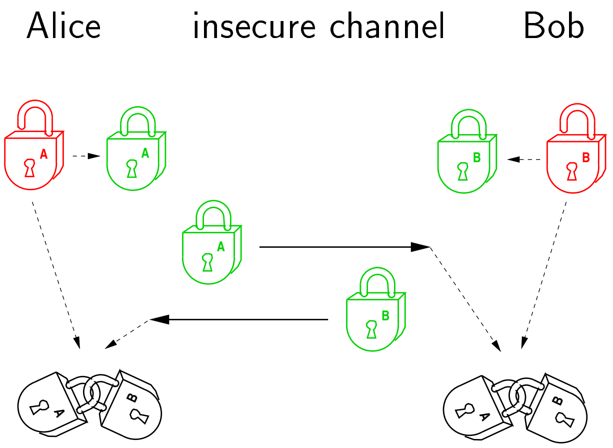

Fig. 1. Mechanical analog of the Diffie-Hellman key agreement protocol. The padlock stands for a one-way function, i.e., a function that is easy to compute but computationally infeasible to invert; it is easy to close the padlock, but infeasible to open it. Alice and Bob each choose a padlock in the open state (i.e., x_A and x_B, respectively) and produce the closed version (i.e., compute y_A and y_B, respectively). The closed locks are exchanged over an insecure channel. Thus both Alice and Bob can produce an identical configuration, namely the two padlocks interlocked, which the adversary cannot generate without breaking one of the locks.

signature scheme, but the first realization for both concepts was proposed by Rivest, Shamir, and Adleman [74] in 1978. Public-key cryptosystems (see the mechanical analog in Figure 2) are closely related to public-key key-agreement. A digital signature scheme allows a signer to generate a signature for a digital document such that everybody (including a judge) can verify such signatures but nobody can forge them.

The basic idea of the RSA system [74] is as follows. To set up the system one publishes a group (with its group operation) but keeps the order t of the group secret. Computing e-th powers (for some publicly known e) is easy, but computing e-th roots without knowledge of t is infeasible. Encryption (signature verification) hence corresponds to raising an element x of the group to a fixed known power e, $y = x^e$, whereas decryption (signature generation) corresponds to computing the e-th root of an element y, which can be achieved by computing $x = y^d$, where d is the multiplicative inverse of e modulo the order of the group: $de \equiv 1 \pmod{t}$. The group proposed in [74] is $\mathbf{Z}_n^*$, where $n = pq$ is the product of two large primes, with p and q being the secret key. The order of the group is $(p-1)(q-1)$ and can be computed if and only if p and q are known.

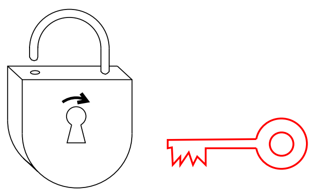

Fig. 2. Mechanical analog of a public-key cryptosystem. Encryption corresponds to closing the padlock, an operation anyone can perform, and decryption corresponds to opening the lock, which requires a secret key.

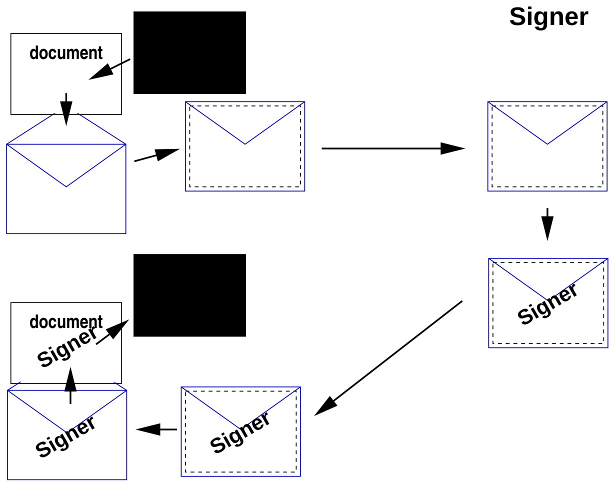

Fig. 3. Mechanical analog of a blind signature scheme. The recipient of the signature (left) puts the document to be signed into an envelope, together with a sheet of carbon paper. The signer (right) signs on the envelope, thus implicitly also signing the document inside.

Blind digital signatures, proposed by Chaum [18] in 1983, allow a signer to digitally sign a message without even knowing (or being able to know) the message that he signs (see Figure 3.) Two applications are anonymous digital cash and certain types of voting schemes. RSA signatures can be issued blindly by multiplying the message m to be signed by r^e, where r is a random blinding factor, and letting the signer sign $\tilde{m} = r^e m$. The signature on $\tilde{m}$ is $\tilde{m}^d = (r^e m)^d = rm^d$. The receiver of this signature can now divide by r to obtain the signature m^d on m.

Zero-knowledge interactive proofs, proposed by Goldwasser, Micali, and Rackoff [37] allow a prover, roughly speaking, to prove a statement (e.g. a mathematical theorem) to a verifier in such a way that the verifier learns nothing he did not know before (except that the statement is true) even if he misbehaves in the protocol. More formally, an interactive proof for a given statement (e.g. a mathematical conjecture) is a protocol between a prover and verifier, at the end of which the verifier makes a binary decision to either accept or reject the proof. If the statement is true, then the verifier accepts with overwhelming probability, and if the statement is false, no prover can make the verifier accept. That the verifier learns nothing means that he can simulate the entire transcript of the protocol, indistinguishable from a real execution of the protocol, without actually talking to the prover. A fundamental result states that every statement which has a conventional proof can also be proved interactively in zero-knowledge [34] (see also [15]).

Secure multi-party computation, another fascinating and paradoxical topic, will be discussed in Section 8.

3 Computational Complexity Theory

One of the main goals of complexity theory (see also Hartmanis' article in the this volume) is to prove lower bounds on the resources (e.g. time and/or space) needed to solve a certain computational problem. Cryptography can therefore be seen as its main "customer". (There are of course plenty of customers for upper bounds, i.e., for efficient algorithms.) While complexity theory has flourished and generated some beautiful and deep results, the needs of the very demanding customer have not even closely been met. The fundamental problems are still wide open.

To begin with, in order to be independent of the details of the computational model, the complexity is not defined for a specific function, but only for an infinite class of functions parameterized with a parameter k (e.g. the input size). The complexity is a function of k. In contrast to such asymptotic definitions, concrete cryptosystems (e.g. DES) are fixed and an asymptotic extension is usually not defined. Moreover, often one distinguishes only between polynomial and super-polynomial complexity, but a finer distinction is of course possible and desirable for concrete security statements and reductions.

Second, for the most part, classical complexity theory deals with worst-case complexity, a concept useless in cryptography where breaking a system must be hard for almost all instances, not just for some of them. Ajtai's celebrated worst-case average-case hardness result [1] was a break-through in the right direction. In cryptography even average-case complexity results are not sufficient: hardness almost everywhere is required.

Third, instead of proving the hardness of finding an exact solution to a computational problem, one would like to prove that even approximating the correct solution is hard. There has been substantial progress in complexity theory in this direction, initiated by Feige et al. [30], based on the theory of probabilistically checkable proofs (e.g. see [45]).

Fourth, the state of the art in lower bound proofs for the computational complexity of any NP-problem is quite miserable, even only for worst-case complexity. The best lower bounds are linear in the input size, which corresponds basically to the time needed to read the input. For example, in the circuit model of computation, a simple counting argument shows that almost all functions from n bits to 1 bit (i.e., $\{0,1\}^n \to \{0,1\}$) have circuit complexity $\Theta(2^n/n)$, but the best lower bound for any concrete function is only $3n$ gates [11]. Proving a lower bound of $4n$, let alone $n \log n$ or n^2, would be a major breakthrough.

Fifth, in 1994 the discussion about the right model of computation as the basis for complexity lower bounds was reopened when Peter Shor [81] showed that certain problems[6] can be solved in polynomial time on a quantum computer. While a quantum computer is still a theoretical model which may perhaps never be implemented at sufficiently large scale to be of practical interest, computer scientists need to rethink the concept of computation. Basically, any process consistent with the laws of physics should be considered a computation, not just steps on an idealized computational model like a Turing machine.

From a cryptographic viewpoint, the most pressing research problem in complexity theory is to prove non-trivial lower bounds, perhaps polynomial[7], perhaps super-polynomial, on the complexity of breaking concrete cryptographic systems, for instance a one-way function.

4 Security Definitions

One of the major research achievements of the past decade or two is the precise definition of the security of many types of cryptographic schemes. Moreover, security proofs of a number of proposed schemes have been given, relative to some well-defined computational intractability assumptions. Some of these results will be mentioned in the corresponding sections below.

[6] in fact the two most relevant cryptographic problems, namely factoring integers and computing discrete logarithms.

[7] Even if P=NP were proved, which would rule out the existence of secure cryptosystems in a classical complexity-theoretic framework, secure cryptography might still be possible within the realm of polynomial computation.

For example, an intuitive security definition for an encryption scheme could be that it is infeasible to compute the plaintext when given the ciphertext, without knowledge of the secret key. However, such a security definition is unsatisfactory for two reasons. First, it should be infeasible to compute *any* information about the plaintext, even to guess a single bit or Boolean predicate of the plaintext with success probability significantly greater than he could before seeing the ciphertext. Second, this should be true even if the adversary can mount various attacks on the cryptosystem, for instance adaptively choose plaintext and receive the corresponding ciphertext, as well as choose ciphertext and receive the corresponding plaintext.

A security definition must specify two things: (1) the adversary's capabilities, i.e., the types of allowed queries to the system under attack and the computational resources, and (2) the task the adversary is required to solve in order to be successful. Strong security definitions are obtained when the adversary is assumed to be as powerful as possible and his task is as easy as possible.

A general methodology for defining the adversary's task is as follows: A purely theoretical *perfect system* is defined which (trivially) satisfies the desired security goals of the type of system under investigation. The adversary's task is defined as the simplest conceivable task, namely only to be able to distinguish between two possible settings: whether it interacts with the real cryptographic system or whether it interacts with the perfect system. If such a distinction is infeasible then, obviously, the real system has all the properties of the perfect system, at least for a computationally bounded observer.[8]

Let us discuss a few examples. The first definition of this kind, proposed by Blum and Micali [10] was for a *pseudo-random bit generator*. Such a generator generates an arbitrarily long sequence of bits, depending on a secret key (often called the seed). It is defined to be secure if no algorithm not knowing the seed can efficiently distinguish the generated sequence from a truly random sequence.

For a block cipher one assumes that the attacker can, using an adaptive strategy, obtain the ciphertexts for chosen plaintexts, and vice versa. The perfect system for a block cipher is a random permutation on the input space, which would trivially be secure. Proving security of a block cipher would mean to prove that it cannot efficiently be distinguished from a random permutation, i.e., that it is a pseudo-random permutation [56].

The security of a *message authentication code (MAC)*[9] can be defined similarly, where the perfect system is a random function from the bit strings of finite length to the range of the MAC (e.g. the 64-bit strings).

A cryptographic scheme (say A) is often constructed from another, typically simpler cryptographic scheme (say B). Denote the combined system by $A(B)$. For instance, the CBC-MAC (e.g. see [4]) is based on a block cipher used in a special feedback mode. Let $\tilde{A}$ and $\tilde{B}$ be the perfect systems corresponding to A and B, respectively. Then the security proof for system $A(B)$, based on the assumption that system B is secure, usually consists of the following two steps (although this is often not made explicit in the literature):

[8] To make this more formal, assume that each of the two systems is selected with probability 1/2. Then they are indistinguishable if for any efficient algorithm the probability of guessing correctly which one is used is only negligibly greater than 1/2. Negligible is usually defined as asymptotically decreasing (as a function of the security parameter) faster the inverse of any polynomial.

[9] A MAC scheme takes a secret key and a message and generates an authenticator value, often called the MAC. Without knowledge of the secret key it should be infeasible to compute the MAC for any message.

1. The proof that $\tilde{A}$ and $A(\tilde{B})$ are indistinguishable, even with infinite computing power, but with at most a polynomial number of queries. This proof is purely probability-theoretic or information-theoretic.
2. The trivial step that $A(\tilde{B})$ and $A(B)$ are computationally indistinguishable. (This follows from that fact that A would be a legitimate distinguisher for $\tilde{B}$ and B, and such a distinguisher cannot exist by assumption.)

The standard notion of security of a signature scheme, introduced in [38], is that after performing an adaptive chosen-message attack the adversary is not able to forge a signature for any message whatsoever not used during the attack.

The strongest security definition for a public-key cryptosystem, proposed by Rackoff and Simon, is semantic security against an adaptive chosen-ciphertext attack [72]. The adversary must specify two plaintexts m_0 and m_1, receives encryptions of m_0 and m_1 in random order, and must guess which encryption belongs to which plaintext. The adversary also has access to a decryption oracle, before and after specifying m_0 and m_1, of course with the (only) restriction that he must not ask for the decryption of any of the encryptions of m_0 and m_1. The cryptosystem is secure if no efficient (e.g. polynomial-time) adversary algorithm can guess correctly in the above game with probability non-negligibly greater than 1/2. Note that this security notion makes sense only for a probabilistic public-key encryption scheme; for a deterministic one (like RSA), the adversary can himself compute the (unique) ciphertext.

A more intricate security notion, introduced by Dolev, Dwork and Naor [29], is *non-malleability*, which captures the notion that it should be infeasible to modify a ciphertext even without knowing the plaintext, in any way that would have a predictable effect on the corresponding plaintext. For example, when given the ciphertext of a bid in an auction, it should be infeasible to come up with the encryption of a smaller bid, with success probability any greater than without being given the ciphertext, even if it is not required to know the plaintext corresponding to the generated ciphertext. This definition shows that secrecy and independence are two separate issues in cryptography. However, it turns out that in the context of an adaptive chosen ciphertext attack, semantic security and non-malleability are equivalent [29].

5 Keyless Cryptographic Functions

Not all cryptographic schemes have a secret key. A *one-way function* f from a domain D to a range R is a fixed function that, roughly speaking, is easy to compute on all inputs but for which it is infeasible, for a y chosen at random from R, to compute an x such that $f(x) = y$. (It may be easy to compute many bits of such an x, but not the entire x.) One-way functions are probably the most fundamental primitive in complexity-based cryptography. Nothing is known about whether one-way functions exist, but intuitively it seems quite easy to design a one-way function. The best provable result known is by Hiltgen [40] who describes a function with linear circuit complexity and with a factor 2 between the complexity of computing the function and its inverse.

A very important class of one-way functions are *cryptographic hash functions*, where $D = \{0,1\}^*$ is the set of all finite-length binary strings and $R = \{0,1\}^k$ for some suitable k, e.g. $k = 128$. Cryptographic hash functions are used for many different purposes, including hashing of messages before signing them, pseudo-random bit generation (e.g. key derivation from master keys), message authentication codes, the conversion of a public-key based challenge-response identification protocol into a signature scheme, by replacing the challenge by a hash value, etc.

The usual definition of cryptographic security of a hash function h is *collision-resistance*, meaning that it should be infeasible to find two distinct inputs $x, x' \in \{0,1\}^*$ such that $h(x) = h(x')$.[10] Most hash functions were designed with collision-resistance as the design goal in mind. But note that each application places a different security requirement on the hash function, and the diversity of applications is enormous. It is an important research topic to determine which security properties are needed in which applications, and which properties can be achieved by a good implementation of a hash function. To avoid such application-dependent investigations, the *random oracle model* was proposed informally in [31] and more formally in [5]. A random oracle is a random function from $D = \{0,1\}^*$ to $R = \{0,1\}^k$, an idealized concept that does not exist in reality. Many system have been proved secure in the random oracle model, but in [16] it was shown that this methodology is inherently problematic: a signature scheme was demonstrated which is secure in the random oracle model but insecure for every implementation of the random oracle by a class of hash functions.

There have been some impressive successes in the cryptanalysis of hash functions (e.g. see [28]).

6 Symmetric Cryptography

6.1 Cipher Design: Towards the Eternally Secure Cipher

A symmetric cryptosystem, called a *cipher*, is an invertible transformation, depending on a secret key, from the plaintext space to the ciphertext space. The first cryptographic algorithm designed to be published widely was the Data Encryption Standard (DES) standardized for U.S. government use in 1974. Although the design criteria for the S-boxes were never published, it has become clear that this is a master piece of cryptographic design, except for its very small key size of only 56 bits, which today, at best, is only marginally secure.

Two other widely used ciphers are the stream cipher RC4, designed by Rivest, and the block cipher IDEA, with a 128-bit key, designed by Lai and Massey [52]. One of the remarkable features of IDEA is that it contains no tables like the S-boxes in DES, and hence the design of tables need not be justified. Instead, IDEA is based purely on the algebraic incompatibility of three group operations on the set of 16-bit strings: bitwise exclusive OR, addition modulo 2^{16}, and multiplication modulo $2^{16}+1$, with 0 interpreted as 2^{16}. Since $2^{16}+1$ is a prime number, this corresponds to the multiplicative group of the finite field $GF(2^{16}+1)$.

Undoubtedly, the third single most important algorithm will become Rijndael [26], designed by Daemen and Rijmen and recently selected among 15 submitted proposals as the proposed new U.S. encryption standard, after an extensive evaluation by the international cryptographic community. The whole submission and evaluation process was one of the most remarkable cryptographic endeavors in the past decade.

The main research goal in symmetric cryptography, as mentioned earlier, would be to devise an encryption algorithm for which the security can be rigorously proven in a reasonable model of computation. This goal seems very far away. However, if one is satisfied without a complete security proof, the state of the art in cipher design is very good: It appears possible today to design a cipher that remains secure for eternity, even assuming that the entire energy and matter

[10] Note that this definition makes sense only when stated for a large class of functions, indexed by a randomly chosen but public parameter (which is given as input to the collision finder algorithm). For a single hash function there always exists a trivial collision finder, namely the algorithm that outputs a particular collision (x, x'). This is the general problem that one cannot define the difficulty of an instance of a problem (e.g. factoring a particular large integer), but only of a problem class (e.g. factoring integers).

of the known universe is thrown at breaking the cipher, provided our present understanding of computation (including quantum computation) remains valid, that $P \neq NP$, and that the algorithm is implemented in a secure environment. Such a cipher would need to have a block size and a key size of a few thousand bits, and it could for instance be based on a combination of many known cipher design principles. In other words, in symmetric cryptography an ultimate victory of the codemakers over the codebreakers seems possible.

6.2 Cryptanalysis

Looking back ten years, perhaps more remarkable than the mentioned new cipher proposals are the contributions to cryptanalysis of symmetric ciphers, which of course can also be interpreted as contributions to the design of secure ciphers. Some impressive generic cryptanalytic techniques are differential cryptanalysis of block ciphers invented by Biham and Shamir [8], linear cryptanalysis proposed by Matsui [58], correlation attacks on stream ciphers proposed by Siegenthaler, generalized by Meier and Staffelbach [63], and subsequently refined by many other authors, and Kocher's timing attacks [50] and differential power analysis [51] of cryptographic implementations (see Section 9). The first two have substantially improved our understanding of what constitutes a secure block cipher and have had a profound impact on all the proposals submitted to the AES competition.

6.3 Security Proofs

In this section we describe some of the results on various types of security proofs for symmetric cryptosystems.

One of the most important result on pseudo-random bit generators[11], due to Impagliazzo, Levin, and Luby [46, 55], is that they can be constructed from any one-way function. This result is remarkable because a scheme satisfying a very strong notion of indistinguishability is constructed from a scheme for which the security notion is merely the infeasibility of computing a value satisfying a certain predicate.

One can prove that the cascade of several ciphers with independent keys is at least as secure as the first cipher, but not necessarily as secure as every cipher in the cascade [57].[12] While this result is less powerful for general cascades than one might hope, it implies that a cascade of commuting ciphers is as secure as the strongest component cipher. Fortunately, additive stream ciphers do commute, a potential advantage over block ciphers.

Luby and Rackoff [56] proved that one can construct pseudo-random permutations from pseudo-random functions [33], which is a function family indistinguishable from a random function. This paper, though much more complicated than necessary in retrospect, has initiated a substantial body of research, including the proof that the CBC-MAC is secure if the underlying block cipher is a pseudo-random function [4].

7 Public-Key Cryptography and Digital Signatures

Public-key cryptography is more fascinating than symmetric-key cryptography because the paradoxical asymmetric functionality of a public-key scheme is much harder to achieve and requires

[11] Their main application is as the key stream generator in a so-called additive stream cipher, in which the pseudo-random bit sequence is added bitwise modulo 2 to the binary plaintext sequence. Stream ciphers are still widely used in military and government applications.

[12] In fact, one can give examples of ciphers [57] which are secure but become insecure when pre-cascaded with another cipher.

substantial mathematical structure, for instance an algebraic group with special properties. Because there is no general design principle for finding such structure, only a limited number of public-key systems have been proposed, in contrast to the large number of proposed symmetric cryptosystems. We mention some of those proposed recently, without having space to describe them.

In addition to new public-key schemes, a number of new concepts were also proposed, for instance fail-stop signatures [71], undeniable signatures [20], and group signature schemes [21].

7.1 New Systems

At present, the discrete logarithm problem in a finite group is the most useful mathematical problem to base public-key cryptography on. A large number of schemes and protocols depend on it. Of particular interest are elliptic curves over finite fields (e.g. see [64]) because for general elliptic curves the fastest known discrete logarithm algorithm are the generic ones that work for any group (see Section 7.3), independently of the representation of the group elements.

In the quest for new public-key systems, several new mathematical principles were proposed as the basis of public-key schemes. The first was McEliece's 1978 proposal [62], based on the hardness of decoding general linear error-correcting codes.

Another important line of research is on multivariate cryptography. The basic idea is that each ciphertext digit is given as a low-degree multi-variate polynomial of the plaintext digits. The public key, consisting of these polynomials, contains a special hidden structure (the secret key) which can be used to decrypt, i.e., solve the system of equations. Several schemes were proposed and some were broken. For instance, Shamir's scheme [78] was broken in [23], and Patarin's hidden fields equations (HFE) scheme [70] was cryptanalysed in [48]. We refer to [49] for a discussion of some such algebraic cryptosystems.

Lattices[13] are playing an important role in cryptanalysis: many cryptosystems have been broken using lattice reduction techniques. Recently, a number of public-key cryptosystems were proposed based on hard problems on lattices, for instance finding the lattice point closest to a given point $P \in \mathbf{R}^n$. Among these are the Ajtai-Dwork scheme [2], the Goldreich-Goldwasser-Halevi scheme [32], and the NTRU scheme [44].

7.2 Cryptanalysis

Only three years after Lenstra's 1987 publication of the elliptic curve factoring algorithm [54], the last decade began with another big bang in factoring: the 1990 discovery of the number field sieve factoring algorithm [53] which drove the asymptotic running time for factoring a k-bit integer from $O(e^{(c+o(1))\sqrt{k \ln k}})$ for some c (for the quadratic sieve) down to $O(e^{(c+o(1))k^{1/3}(\ln k)^{2/3}})$ for some c. This algorithm was used to establish the impressive factoring record to date [17], RSA Inc.'s 512-bit factoring challenge. It can also be adapted to compute discrete logarithms modulo a prime number. In the past decade substantial progress in computational number theory was made, not only in factoring.

Some remarkable results in the cryptanalysis of public-key schemes were the sub-exponential discrete logarithm algorithm for super-singular elliptic curves due to Menezes et al. [66], the breaking by Vaudenay [86] of the Chor-Rivest cryptosystem (at least of the proposed challenge

[13] An n-dimensional lattice is given by a set of n basis vectors in $\mathbf{R}^n$, and consists of all integer linear combinations of the basis vectors.

instance), the various attacks against multivariate cryptosystems mentioned above, Bleichenbacher's attack on the RSA encryption standard PKCS #1 [9], and the Phong-Stern [69] attack against the Ajtai-Dwork scheme. This list is not complete.

7.3 Security Proofs

As mentioned earlier, Dolev et al. introduced non-malleability as a concept and also proposed a provably non-malleable but not practical public-key cryptosystem (based on an intractability assumption).

Shoup [82] proved that every generic discrete logarithm algorithm (i.e., one that works in any group) has complexity on the order of $\sqrt{q}$, where q is the largest prime factor of the group order. In [60] it is proved that one can construct groups for which breaking the Diffie-Hellman protocol is provably as hard as computing discrete logarithms, and this equivalence holds for any group if a plausible number-theoretic conjecture holds.

The Cramer-Shoup cryptosystem [25] was the first practical public-key scheme provably secure against adaptive chosen ciphertext attacks, based on the Decision-Diffie-Hellman assumption. This assumption states that it is infeasible in certain groups of large prime order to distinguish a Diffie-Hellman triple (g^a, g^b, g^{ab}) from a random triple (say (g^a, g^b, g^c)) of group elements.

In the past years, researchers have devised practical schemes provably secure against more powerful adversaries, but often the price was a stronger intractability assumption. Another such assumption is the so-called strong RSA assumption, namely that it is infeasible, for a given integer n with unknown factorization and a given random y, to find x and e such that $x^e \equiv y \pmod{n}$. Note that the regular RSA assumption is a weaker assumption because e is also fixed and hence there is less freedom in finding a solution. In the future we will have to prove the security of practical schemes for much weaker assumptions.

For several public-key schemes it was proved that computing certain bits of the secret (the plaintext or the shared key in case of the Diffie-Hellman protocol) is as hard as computing the entire value [3, 13, 39].

8 Cryptographic Protocols

Cryptographic protocols, often based on public-key functionality, are among the most delicate and fascinating topics in cryptography. In the past decade there has been an extensive research activity on various types of protocols, in particular on interactive proofs and zero-knowledge protocols (see Section 2), as well as on secure multi-party computation. For lack of space, we only discuss the latter.

Secure multi-party computation, introduced by Yao [87], enables a set of n players to compute an arbitrary agreed function of their private inputs. The correctness of the outputs and the secrecy of the players' inputs is guaranteed, even if any t of the players are corrupted by an adversary and misbehave maliciously. Surprisingly, this task is possible if $t < n/2$ [35], based on cryptographic intractability assumptions, and without such assumptions if $t < n/3$ [7, 19], assuming the players are connected by pairwise secure channels.

After these fundamental results were published in the late eighties, multi-party computation has seen a revival in the late nineties. Some results are mentioned briefly. Efficient protocols for concrete problems were developed, for instance for voting [24, 43], shared RSA key generation [12], and threshold public-key cryptosystems and signature schemes (e.g. see [83] and the references therein). Threshold adversaries were generalized to general adversary structures [41], and the efficiency of protocols that work for any function were improved from n^6 to n^3 per multiplication in the function specification [42].

9 The Role of Physics

Algorithms, both for using and breaking cryptographic systems, are traditionally modeled as mathematical objects, and therefore cryptography and cryptanalysis are traditionally seen as mathematical disciplines. However, all processes in the real world, including computation and communication, are physical processes involving noise, quantum effects, and other uncertainty factors. This fact can be used both for the benefit and detriment of cryptography.

On the negative side, two developments should be mentioned. The first is quantum computing, leading potentially to efficient methods for factoring large numbers and computing discrete logarithms [81] (see also Vitanyi's article in this volume). The second is the exploitation (by the cryptanalyst) of different types of side information channels which are available because a cryptosystem is implemented and embedded in a physical environment (e.g. a smart-card). Examples of such channels are the time required for a certain computation [50], the power consumption as a function of what is computed [51], electro-magnetic radiation, or the behavior of a device when faults are provoked, e.g. by setting the power level outside of the specifications, by physical stress, or by applying electro-magnetic fields.

Designing tamper-resistant devices with only low-capacity side channels is a core issue in security engineering. Cryptographic research must propose adequate models of side channels and resistant implementation techniques, for instance by masking key values or reordering the sequence of instructions. In addition, one should try to devise algorithms whose implementations can be inherently less vulnerable to side-channel attacks.

On the positive side, one can exploit the fact that an adversary does not know the exact state of a physical system. A first possible source of uncertainty for the adversary is the inherent impossibility of measuring quantum states exactly. For instance, one can measure at most one bit of information about the polarization of a photon, which is a continuous quantity. Quantum cryptography (e.g. see [6]) is the discipline that exploits this fact.

A second source of uncertainty for the adversary is the noise in communication channels. Cryptography is usually used in a context where error-free communication channels (e.g. TCP connections) are available. In such a model, Shannon's theorem [80] applies which states that information-theoretic secrecy is possible only if, roughly speaking, the key is as long as the message. It was shown in [59] that by combining cryptographic coding and error-control coding, perfect information-theoretic secrecy can be achieved by public discussion between two parties not sharing a secret key initially, by exploiting information available to them as well as (even more reliably) to the adversary.

10 Applications and Politics

The potential of public-key cryptography for large scale applications, where classical key management is infeasible, was recognized immediately after its discovery. But neither the technology for implementing it efficiently, nor the market was there. Only around 1990, and in particular due to the success of the Internet after the mid-nineties, the commercial market for cryptography began to develop. But the market is still far from where it should be given the very strong current interest and need for cryptography in Internet-based applications.

Why is cryptography not yet built into every software product where this would be appropriate? The single most important reason is probably the U.S. export control regulation which, until very recently, did not allow U.S. companies to export cryptographic systems with keys longer than 40 bits, which is quite insecure.[14] One may ask why it makes sense to forbid the export of the source code of DES, as an example, when implementing DES is trivial. But the effect of the U.S. export restrictions was substantial: they prevented standardization from happening because U.S. companies had little interest in defining standards to allow the rest of the world to penetrate the growing global security market. Hence the National Security Agency's export control campaign can be seen as quite successful in reaching its goal, namely keeping secure systems away from those the United States wants to be able to eavesdrop on. Cryptography will continue to be a political topic.

The first global success story of a cryptographic implementation was Phil Zimmermann's Pretty Good Privacy (PGP) e-mail encryption program [88], which since its release on the Internet in the early nineties was used by millions of users. The most remarkable feature of PGP is its key management concept, which allowed the Internet community to build a rudimentary public-key infrastructure (called by Zimmermann the "grass-roots approach") long before the term PKI became a buzz word in the commercial world.

Another success story is the Secure Socket Layer (SSL) protocol, designed by Netscape, which is now used in a large number of e-business applications. One of the most useful applications, solutions for which are studied theoretically but are still not surfacing in the marketplace, is a global secure payment system. The spectrum of technical possibilities is wide, ranging from secure privacy-enhanced funds-transfer systems to fully anonymous on-line or off-line digital cash (e.g. see [14]). Which types of system will prevail will depend more on the market forces than on technical issues. In fact, perhaps the greatest commercial failure of cryptographic solutions can be found in the payment arena: Digicash went bankrupt despite its technically innovative products. It is difficult to speculate about how long it will take until normal cash has more or less disappeared in the modern world, but I expect this to happen within the next 20 years.

The main requirement for cryptographic application to become universal is a global interoperable public-key infrastructure (PKI). There exist many initiatives to build a PKI in certain application contexts, but it may take a number of years before a global PKI is available. In fact, many issues are not yet well understood.

11 The Future of Cryptography

The crucial components of the emerging global information infrastructure will have to be systems that operate reliably and securely in a potentially highly adversarial environment. The dependence on information systems will increase strongly. Security breaches and system failures will not only be a nuisance and a cost factor, but will be intolerable, possibly major disasters.

Information security engineering involves many different disciplines, including most prominently cryptography. There is no doubt that cryptography will be a crucial discipline for the emerging information society. While security was in the past often seen as an expensive obstacle for IT projects, thanks to cryptography it is more and more perceived as an enabling technology, laying the basis for electronic commerce.

[14] This policy had some quite strange consequences. For example, Pretty Good Privacy (PGP) [88], a product of Network Associates, Inc., was exported from the U.S. in full compliance with the law, by publishing the source code as a book, legally exporting the book, and scanning it in Switzerland to reproduce a fully compatible version of PGP.

There is also a backside. The information society will depend strongly on the security of cryptographic schemes, in particular digital signature schemes. Cryptanalysis is therefore a potential source of major disasters. For instance, the discovery (and publication) of a fast factoring algorithm would be a catastrophe today, but in ten years it might even cause a serious problem for the world economy, unless utmost care is taken in the design of future security systems, in the choice of cryptographic algorithms, and in legislation. Very cautious cryptographic design (for instance the parallel use of several signature schemes) is of paramount importance, especially because not only future transactions, but the validity of all existing signatures is in danger. Moreover, fall-back scenarios for the case that a cryptosystem is broken must be part of cryptographic design.

A major future research theme is to weaken the assumptions on which security proofs are based, and to make these assumptions explicit:

- *Computational intractability assumptions.* Given the lack of full-fledged complexity-theoretic security proofs we need to continue the nice research of the past decade on weakening these assumptions.
- *Trust assumptions.* As the important paradigm of secure multi-party computation shows, trust in one particular entity, person, software or hardware component can be reduced by distributed protocols involving several parties, only some of which need to be trusted.
- *Physical assumptions.* Computation and communication are physical processes involving noise, quantum effects, and other uncertainty factors. This fact can be used both for the benefit and detriment of cryptographic applications.

Acknowledgments

I would like to thank Cynthia Dwork, Ron Rivest, and Jacques Stern for interesting discussions on the past and the future of cryptography, and Arjen Lenstra and Ron Rivest for valuable feedback on an earlier version of this paper.

References

1. M. Ajtai, Generating hard instances of lattice problems, *Proc. 28th ACM Symposium on the Theory of Computing (STOC)*, pp. 99–108, 1996.
2. M. Ajtai and C. Dwork, A public-key cryptosystem with worst-case/average-case equivalence, *Proc. 29th ACM Symposium on the Theory of Computing (STOC)*, pp. 284–293, 1997.
3. W. Alexi, B. Chor, O. Goldreich, and C. Schnorr, RSA and Rabin functions: certain parts are as hard as the whole, *SIAM Journal on Computing*, vol. 17, no. 2, pp. 194–209, 1988.
4. M. Bellare, J. Kilian, and P. Rogaway, The security of the cipher block chaining message authentication code, *Advances in Cryptology - CRYPTO '94*, Lecture Notes in Computer Science, vol. 839, pp. 455–469, Springer-Verlag, 1995.
5. M. Bellare and P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, *Proc. First ACM Conference on Computer and Communication Security*, pp. 62–73. Association for Computing Machinery, 1993.
6. C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental quantum cryptography," *Journal of Cryptology*, vol. 5, no. 1, pp. 3–28, Springer-Verlag, 1992.
7. M. Ben-Or, S. Goldwasser, and A. Wigderson, Completeness theorems for non-cryptographic fault-tolerant distributed computation, *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pp. 1–10, 1988.
8. E. Biham and A. Shamir, Differential cryptanalysis of the Data Encryption Standard, New York: Springer-Verlag, 1993.
9. D. Bleichenbacher, Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1, *Advances in Cryptology — CRYPTO '98*, Lecture Notes in Computer Science, vol. 1462, pp. 1–12, Springer-Verlag, 1998.

10. M. Blum and S. Micali, How to generate cryptographically strong sequences of pseudo-random bits, *SIAM Journal on Computing*, vol. 13, no. 4, pp. 850–864, 1984.
11. N. Blum, A boolean function requiring $3n$ network size, *Theoretical Computer Science*, vol. 28, pp. 337–345, 1984.
12. D. Boneh and M. Franklin, Efficient generation of shared RSA keys, *Advances in Cryptology - CRYPTO '97*, Lecture Notes in Computer Science, vol. 1294, pp. 425–439, Springer-Verlag, 1997.
13. D. Boneh and R. Venkatesan, Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes, *Advances in Cryptology - CRYPTO '96*, Lecture Notes in Computer Science, vol. 1109, pp. 129–142, Springer-Verlag, 1996.
14. S. Brands, Electronic cash systems based on the representation problem in groups of prime order, *Advances in Cryptology – CRYPTO '93*, Lecture Notes in Computer Science, vol. 773, pp. 302–318, Springer-Verlag, 1994.
15. G. Brassard, D. Chaum, and C. Crépeau, Minimum disclosure proofs of knowledge, *Journal of Computer and System Sciences*, vol. 37, no. 2, pp. 156–189, 1988.
16. R. Canetti, O. Goldreich, and S. Halevi, The random oracle methodology, revisited, *Proc. 30th ACM Symposium on the Theory of Computing (STOC)*, pp. 209–218, 1998.
17. S. Cavallar et al., Factorization of a 512-bit RSA modulus, *Advances in Cryptology - EUROCRYPT 2000*, Lecture Notes in Computer Science, vol. 1807, pp. 1–18, Springer-Verlag, 2000.
18. D. Chaum, Blind signature systems, *Advances in Cryptology — CRYPTO '83*, p. 153, Plenum Press, 1984.
19. D. Chaum, C. Crépeau, and I. Damgård, Multiparty unconditionally secure protocols, *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pp. 11–19, 1988.
20. D. Chaum and H. van Antwerpen, Undeniable signatures, *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science, vol. 435, pp. 212–216, Springer-Verlag, 1990.
21. D. Chaum and E. van Heyst, Group signatures, *Advances in Cryptology — EUROCRYPT '91*, Lecture Notes in Computer Science, vol. 547, pp. 257–265, Springer-Verlag, 1991.
22. D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter, Low-exponent RSA with related messages, *Advances in Cryptology – EUROCRYPT '96*, Lecture Notes in Computer Science, vol. 1070, pp. 1–9, Springer-Verlag, 1996.
23. D. Coppersmith, J. Stern, and S. Vaudenay, The security of the birational permutation signature schemes, *Journal of Cryptology*, vol. 10, no. 3, pp. 207–221, 1997.
24. R. Cramer, R. Gennaro, and B. Schoenmakers, A secure and optimally efficient multi-authority election scheme, *European Transactions on Telecommunications*, vol. 8, pp. 481–489, Sept. 1997.
25. R. Cramer and V. Shoup, A practical public-key cryptosystem provably secure against adaptive chosen ciphertext attack, *Advances in Cryptology — CRYPTO '98*, Lecture Notes in Computer Sc., vol. 1462, pp. 13–25, Springer-Verlag, 1998.
26. J. Daemen and V. Rijmen, The Rijndael block cipher – AES Proposal, see http:/csrc.nist.gov/encryption/aes/rijndael/.
27. W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.
28. H. Dobbertin, Cryptanalysis of MD4, *Journal of Cryptology*, vol. 11, no. 4, pp. 253–271, 1998.
29. D. Dolev, C. Dwork, and M. Naor, Non-malleable cryptography, *Proc. 23rd ACM Symposium on the Theory of Computing (STOC)*, pp. 542–552, 1991. (Also to appear in *SIAM Journal on Computing*.)
30. U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, Approximating clique is almost NP-complete, *Proc. 32nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 2–12, IEEE Press, 1991.
31. A. Fiat and A. Shamir, How to prove yourself: practical solution to identification and signature problems, *Advances in Cryptology - CRYPTO '86*, Lecture Notes in Computer Science, vol. 263, pp. 186–194, Springer-Verlag, 1987.
32. O. Goldreich, S. Goldwasser, and S. Halevi, Public-key cryptosystems from lattice reduction problems, *Advances in Cryptology - CRYPTO '97*, Lecture Notes in Computer Science, vol. 1294, pp. 112–131, Springer-Verlag, 1997.
33. O. Goldreich, S. Goldwasser, and S. Micali, How to construct random functions, *Journal of the ACM*, vol. 33, no. 4, pp. 210–217, 1986.
34. O. Goldreich, S. Micali, and A. Wigderson, How to prove all NP statements in zero-knowledge, and a methodology of cryptographic protocol design, *Advances in Cryptology - CRYPTO '86*, Lecture Notes in Computer Science, vol. 263, pp. 171–185, Springer-Verlag, 1987.
35. O. Goldreich, S. Micali, and A. Wigderson, How to play any mental game – a completeness theorem for protocols with honest majority, *Proc. 19th ACM Symposium on the Theory of Computing (STOC)*, pp. 218–229, 1987.

36. S. Goldwasser and S. Micali, Probabilistic encryption, *Journal of Computer and System Sciences*, vol. 28, pp. 270–299, 1984.
37. S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems, *SIAM Journal on Computing*, vol. 18, pp. 186–208, 1989.
38. S. Goldwasser, S. Micali, and R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, *SIAM Journal on Computing*, vol. 17, no. 2, pp. 281–308, 1988.
39. J. Håstad and M. Näslund, The security of individual RSA bits, *Proc. 39th IEEE Symposium on the Foundations of Computer Science (FOCS)*, pp. 510–519, IEEE Press, 1998.
40. A. Hiltgen, Constructions of feebly-one-way families of permutations, *Advances in Cryptology – Auscrypt '92*, Lecture Notes in Computer Science, pp. 422–434, Springer-Verlag, 1993.
41. M. Hirt and U. Maurer, Player simulation and general adversary structures in perfect multi-party computation, *Journal of Cryptology*, vol. 13, no. 1, pp. 31–60, 2000.
42. M. Hirt, U. Maurer, and B. Przydatek, Efficient secure multi-party computation, to appear in *Advances in Cryptology - Asiacrypt 2000*, Lecture Notes in Computer Science, Springer-Verlag, 2000.
43. M. Hirt and K. Sako, Efficient receipt-free voting based on homomorphic encryption, *Advances in Cryptology - EUROCRYPT 2000*, Lecture Notes in Computer Science, vol. 1807, pp. 539–556, Springer-Verlag, 2000.
44. J. Hoffstein, J. Pipher, and J. Silverman, NTRU: A ring based public key cryptosystem, *Algorithmic Number Theory (ANTS III)*, Lecture Notes in Computer Science, vol. 1423, pp. 267–288, Springer-Verlag, 1998.
45. S. Hougardy, H. J. Prömel, and A. Steger, Probabilistically checkable proofs and their consequences for approximation algorithms, *Discrete Mathematics*, vol. 9, pp. 175–223, North Holland, 1995.
46. R. Impagliazzo, L. Levin, and M. Luby, Pseudo-random generation from one-way functions, *Proc. 21st ACM Symposium on the Theory of Computing (STOC)*, pp. 12–24, 1989.
47. D. Kahn, *The code breakers, the story of secret writing*, MacMillan, New York, 1967.
48. A. Kipnis and A. Shamir, Cryptanalysis of the HFE public key cryptosystem by relinearization, *Advances in Cryptology - CRYPTO '99*, Lecture Notes in Computer Science, vol. 1666, pp. 19–30, Springer-Verlag, 1999.
49. N. Koblitz, *Algebraic Aspects of Cryptography*, Berlin: Springer-Verlag, 1998.
50. P. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, *Advances in Cryptology - CRYPTO '96*, Lecture Notes in Computer Science, vol. 1109, pp. 104–113, Springer-Verlag, 1996.
51. P. Kocher, J. Jaffe, and B. Jun, Differential power analysis, *Advances in Cryptology - CRYPTO '99*, Lecture Notes in Computer Science, vol. 1666, pp. 388–397, Springer-Verlag, 1999.
52. X. Lai and J. L. Massey, A proposal for a new block encryption standard, *Advances in Cryptology – EUROCRYPT '90*, Lecture Notes in Computer Science, vol. 473, pp. 389–404, Springer-Verlag, 1991.
53. A. K. Lenstra, H.W. Lenstra, M.S. Manasse, and J.M. Pollard, The number field sieve, *Proc. 22nd ACM Symposium on Theory of Computing*, pp. 564–572, 1990.
54. H. W. Lenstra, Jr., Factoring integers with elliptic curves, *Annals of Mathematics*, vol. 126, pp. 649–673, 1987.
55. M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton University Press, 1996.
56. M. Luby and C. Rackoff, How to construct pseudorandom permutations from pseudorandom functions, *SIAM Journal on Computing*, vol. 17, no. 2, pp. 373–386, 1988.
57. J. L. Massey and U. Maurer, Cascade ciphers: the importance of being first, *Journal of Cryptology*, vol. 6, no. 1, pp. 55–61, 1993.
58. M. Matsui, Linear Cryptanalysis Method for DES Cipher, *Advances in Cryptology – EUROCRYPT '93*, Lecture Notes in Computer Science, vol. 765, pp. 386–397, Springer-Verlag, 1994.
59. U. Maurer, Secret key agreement by public discussion from common information, *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, 1993.
60. U. Maurer and S. Wolf, On the complexity of breaking the Diffie-Hellman protocol, *SIAM Journal on Computing*, vol. 28, pp. 1689–1721, 1999.
61. K. S. McCurley, The discrete logarithm problem, in *Cryptology and computational number theory*, C. Pomerance (Ed.), Proc. of Symp. in Applied Math., vol. 42, pp. 49–74, American Mathematical Society, 1990.
62. R. J. McEliece, A public-key cryptosystem based on algebraic coding theory, DSN progress report 42–44, Jet Propulsion Laboratory, Pasadena, 1978.
63. W. Meier and O. Staffelbach, Fast correlation attacks on stream ciphers, *Journal of Cryptology*, vol. 1, no. 3, pp. 159–176, 1989.
64. A. J. Menezes, *Elliptic curve public key cryptosystems*, Kluwer Academic Publishers, 1993.
65. A. J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, Boca Raton: CRC Press, 1997.
66. A. J. Menezes, T. Okamoto, and S.A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *Proc. 23rd ACM Symposium on the Theory of Computing (STOC)*, pp. 80–89, 1991.

67. R. Merkle, Secure communication over insecure channels, *Communications of the ACM*, vol. 21, no. 4, pp. 294–299, 1978.
68. R. Merkle, A certified digital signature, *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science, vol. 435, pp. 218–238, Springer-Verlag, 1990.
69. P. Nguyen and J. Stern, Cryptanalysis of the Ajtai-Dwork cryptosystem, *Advances in Cryptology — CRYPTO '98*, Lecture Notes in Computer Science, vol. 1462, pp. 243–256, Springer-Verlag, 1998.
70. J. Patarin, Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms, *Advances in Cryptology — EUROCRYPT '96*, Lecture Notes in Computer Science, vol. 1070, pp. 33–48, Springer-Verlag, 1996.
71. B. Pfitzmann and M. Waidner, Fail-stop signatures and their application, *Proc. SECURICOM '91*, pp. 145–160, 1991.
72. C. Rackoff and D. Simon, Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack, *Advances in Cryptology — CRYPTO '91*, Lecture Notes in Computer Science, vol. 576, pp. 377–391, Springer-Verlag, 1991.
73. R. L. Rivest, Cryptography, Chapter 13 in *Handbook of Theoretical Computer Science*, (J. van Leeuwen, ed.), MIT Press, 1990.
74. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.
75. B. Schneier, *Applied Cryptography*, Wiley, 2nd edition, 1996.
76. C. P. Schnorr, Efficient signature generation for smart cards, *Journal of Cryptology*, vol. 4, no. 3, pp. 239–252, 1991.
77. A. Shamir, IP = PSPACE, *Proc. 31st Annual Symposium on Foundations of Computer Science (FOCS)*, vol. I, pp. 11–15, IEEE Press, 1990.
78. A. Shamir, Efficient signature schemes based on birational permutations, *Advances in Cryptology – CRYPTO '93*, Lecture Notes in Computer Science, vol. 773, pp. 1–12, Springer-Verlag, 1994.
79. C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.
80. C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.
81. P. W. Shor, Algorithms for quantum computation: discrete log and factoring, *Proc. 35th IEEE Symposium on the Foundations of Computer Science (FOCS)*, pp. 124–134, IEEE Press, 1994.
82. V. Shoup, Lower bounds for discrete logarithms and related problems, *Advances in Cryptology - EUROCRYPT '97*, Lecture Notes in Computer Science, vol. 1233, pp. 256–266, Springer-Verlag, 1997.
83. V. Shoup, Practical threshold signatures, *Advances in Cryptology - EUROCRYPT 2000*, Lecture Notes in Computer Science, vol. 1807, pp. 207–220, Springer-Verlag, 2000.
84. S. Singh, *The Code Book*, Fourth Estate, London, 1999.
85. D.R. Stinson, *Cryptography – Theory and Practice*, CRC Press, 1995.
86. S. Vaudenay, Cryptanalysis of the Chor-Rivest cryptosystem, *Advances in Cryptology — CRYPTO '98*, Lecture Notes in Computer Science, vol. 1462, pp. 243–256, Springer-Verlag, 1998.
87. A. C. Yao, Protocols for secure computations, *Proc. 23rd IEEE Symposium on the Foundations of Computer Science (FOCS)*, pp. 160–164, IEEE Press, 1982.
88. P. R. Zimmermann, *The Official PGP User's Guide*, MIT Press, 1995.