TEXTBOOKS IN MATHEMATICS

Cryptography

Theory and Practice

FOURTH EDITION

Douglas R. Stinson
Maura B. Paterson

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Cryptography

Theory and Practice

Fourth Edition

Textbooks in Mathematics

Series editors:
Al Boggess and Ken Rosen

MATHEMATICAL MODELING FOR BUSINESS ANALYTICS
William P. Fox

ELEMENTARY LINEAR ALGEBRA
James R. Kirkwood and Bessie H. Kirkwood

APPLIED FUNCTIONAL ANALYSIS, THIRD EDITION
J. Tinsley Oden and Leszek Demkowicz

AN INTRODUCTION TO NUMBER THEORY WITH CRYPTOGRAPHY, SECOND EDITION
James R. Kraft and Lawrence Washington

MATHEMATICAL MODELING: BRANCHING BEYOND CALCULUS
Crista Arangala, Nicolas S. Luke and Karen A. Yokley

ELEMENTARY DIFFERENTIAL EQUATIONS, SECOND EDITION
Charles Roberts

ELEMENTARY INTRODUCTION TO THE LEBESGUE INTEGRAL
Steven G. Krantz

LINEAR METHODS FOR THE LIBERAL ARTS
David Hecker and Stephen Andrilli

CRYPTOGRAPHY: THEORY AND PRACTICE, FOURTH EDITION
Douglas R. Stinson and Maura B. Paterson

DISCRETE MATHEMATICS WITH DUCKS, SECOND EDITION
Sarah-Marie Belcastro

BUSINESS PROCESS MODELING, SIMULATION AND DESIGN, THIRD EDITION
Manual Laguna and Johan Marklund

GRAPH THEORY AND ITS APPLICATIONS, THIRD EDITION
Jonathan L. Gross, Jay Yellen and Mark Anderson

Cryptography

Theory and Practice

Fourth Edition

Douglas R. Stinson
Maura B. Paterson

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20180724

International Standard Book Number-13: 978-1-1381-9701-5 (Hardback)

Library of Congress Cataloging-in-Publication Data

Names: Stinson, Douglas R. (Douglas Robert), 1956- author. | Paterson, Maura B., author.
Title: Cryptography : theory and practice / Douglas R. Stinson and Maura B. Paterson.
Description: Fourth edition. | Boca Raton : CRC Press, Taylor & Francis Group, 2018.
Identifiers: LCCN 2018018724 | ISBN 9781138197015
Subjects: LCSH: Coding theory. | Cryptography.
Classification: LCC QA268 .S75 2018 | DDC 005.8/2--dc23
LC record available at https://lccn.loc.gov/2018018724

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

To my children, Michela and Aiden
DRS

To my father, Hamish
MBP

Contents

Preface

The first edition of this book was published in 1995. The objective at that time was to produce a general textbook that treated all the essential core areas of cryptography, as well as a selection of more advanced topics. More recently, a second edition was published in 2002 and the third edition appeared in 2006.

There have been many exciting advances in cryptography since the publication of the first edition of this book 23 years ago. At the same time, many of the "core" areas of cryptography that were important then are still relevant now—providing a strong grounding in the fundamentals remains a primary goal of this book. Many decisions had to be made in terms of which older topics to retain and which new subjects should be incorporated into the book. Our choices were guided by criteria such as the relevance to practical applications of cryptography as well as the influence of new approaches and techniques to the design and analysis of cryptographic protocols. In many cases, this involved studying cutting-edge research and attempting to present it in an accessible manner suitable for presentation in the classroom.

In light of the above, the basic core material of secret-key and public-key cryptography is treated in a similar fashion as in previous editions. However, there are many topics that have been added to this edition, the most important being the following:

- There is a brand new chapter on the exciting, emerging area of post-quantum cryptography, which covers the most important cryptosystems that are designed to provide security against attacks by quantum computers (Chapter 9).
- A new high-level, nontechnical overview of the goals and tools of cryptography has been added (Chapter 1).
- A new mathematical appendix is included, which summarizes definitions and main results on number theory and algebra that are used throughout the book. This provides a quick way to reference any mathematical terms or theorems that a reader might wish to find (Appendix A).
- An expanded treatment of stream ciphers is provided, including common design techniques along with a description of the popular stream cipher known as *Trivium*.
- The book now presents additional interesting attacks on cryptosystems, including:

  - padding oracle attack
  - correlation attacks and algebraic attacks on stream ciphers
  - attack on the *DUAL-EC* random bit generator that makes use of a trapdoor.

- A treatment of the sponge construction for hash functions and its use in the new *SHA-3* hash standard is provided. This is a significant new approach to the design of hash functions.
- Methods of key distribution in sensor networks are described.
- There is a section on the basics of visual cryptography. This allows a secure method to split a secret visual message into pieces (shares) that can later be combined to reconstruct the secret.
- The fundamental techniques of cryptocurrencies, as used in BITCOIN and blockchain, are described.
- We explain the basics of the new cryptographic methods employed in messaging protocols such as *Signal*. This includes topics such as deniability and Diffie-Hellman key ratcheting.

We hope that this book can be used in a variety of courses. An introductory undergraduate level course could be based on a selection of material from the first eight chapters. We should point out that, in several chapters, the later sections can be considered to be more advanced than earlier sections. These sections could provide material for graduate courses or for self-study. Material in later chapters can also be included in an introductory or follow-up course, depending on the interests of the instructor.

Cryptography is a broad subject, and it requires knowledge of several areas of mathematics, including number theory, groups, rings and fields, linear algebra, probability and information theory. As well, some familiarity with computational complexity, algorithms, and NP-completeness theory is useful. In our opinion, it is the breadth of mathematical background required that often creates difficulty for students studying cryptography for the first time. With this in mind, we have maintained the mathematical presentation from previous editions. One basic guiding principle is that understanding relevant mathematics is essential to the comprehension of the various cryptographic schemes and topics. At the same time, we try to avoid unnecessarily advanced mathematical techniques—we provide the essentials, but we do not overload the reader with superfluous mathematical concepts.

The following features are common to all editions of this book:

- Mathematical background is provided where it is needed, in a "just-in-time" fashion.
- Informal descriptions of the cryptosystems are given along with more precise pseudo-code descriptions.

- Numerical examples are presented to illustrate the workings of most of the algorithms described in the book.
- The mathematical underpinnings of the algorithms and cryptosystems are explained carefully and rigorously.
- Numerous exercises are included, some of them quite challenging.

We have received useful feedback from various people on the content of this book as we prepared this new edition. In particular, we would like to thank Colleen Swanson for many helpful comments and suggestions. Several anonymous reviewers provided useful suggestions, and we also appreciate comments from Steven Galbraith and Jalaj Upadhyay. Finally, we thank Roberto De Prisco, who prepared the examples of shares in a visual threshold scheme that are included in Chapter 11.

Douglas R. Stinson
Maura B. Paterson

Chapter 1

Introduction to Cryptography

In this chapter, we present a brief overview of the kinds of problems studied in cryptography and the techniques used to solve them. These problems and the cryptographic tools that are employed in their solution are discussed in more detail and rigor in the rest of this book. This introduction may serve to provide an informal, non-technical, non-mathematical summary of the topics to be addressed. As such, it can be considered to be optional reading.

1.1 Cryptosystems and Basic Cryptographic Tools

In this section, we discuss basic notions relating to encryption. This includes secret-key and public-key cryptography, block and stream ciphers, and hybrid cryptography.

1.1.1 Secret-key Cryptosystems

Cryptography has been used for thousands of years to help to provide confidential communications between mutually trusted parties. In its most basic form, two people, often denoted as ***Alice*** and ***Bob***, have agreed on a particular ***secret key***. At some later time, Alice may wish to send a secret message to Bob (or Bob might want to send a message to Alice). The key is used to transform the original message (which is usually termed the ***plaintext***) into a scrambled form that is unintelligible to anyone who does not possess the key. This process is called ***encryption*** and the scrambled message is called the ***ciphertext***. When Bob receives the ciphertext, he can use the key to transform the ciphertext back into the original plaintext; this is the ***decryption*** process. A ***cryptosystem*** constitutes a complete specification of the keys and how they are used to encrypt and decrypt information.

Various types of cryptosystems of increasing sophistication have been used for many purposes throughout history. Important applications have included sensitive communications between political leaders and/or royalty, military maneuvers, etc. However, with the development of the internet and applications such as electronic commerce, many new diverse applications have emerged. These include scenarios such as encryption of passwords, credit card numbers, email, documents, files, and digital media.

It should also be mentioned that cryptographic techniques are also widely used to protect stored data in addition to data that is transmitted from one party to another. For example, users may wish to encrypt data stored on laptops, on external hard disks, in the cloud, in databases, etc. Additionally, it might be useful to be able to perform computations on encrypted data (without first decrypting the data).

The development and deployment of a cryptosystem must address the issue of security. Traditionally, the threat that cryptography addressed was that of an eavesdropping adversary who might intercept the ciphertext and attempt to decrypt it. If the adversary happens to possess the key, then there is nothing that can be done. Thus the main security consideration involves an adversary who does not possess the key, who is still trying to decrypt the ciphertext. The techniques used by the adversary to attempt to "break" the cryptosystem are termed ***cryptanalysis***. The most obvious type of cryptanalysis is to try to guess the key. An attack wherein the adversary tries to decrypt the ciphertext with every possible key in turn is termed an ***exhaustive key search***. When the adversary tries the correct key, the plaintext will be found, but when any other key is used, the "decrypted" ciphertext will likely be random gibberish. So an obvious first step in designing a secure cryptosystem is to specify a very large number of possible keys, so many that the adversary will not be able to test them all in any reasonable amount of time.

The model of cryptography described above is usually called ***secret-key cryptography***. This indicates that there is one secret key, which is known to both Alice and Bob. That is, the key is a "secret" that is known to two parties. This key is employed both to encrypt plaintexts and to decrypt ciphertexts. The actual encryption and decryption functions are thus inverses of each other. Some basic secret-key cryptosystems are introduced and analyzed with respect to different security notions in Chapters 2 and 3.

The drawback of secret-key cryptography is that Alice and Bob must somehow be able to agree on the secret key ahead of time (before they want to send any messages to each other). This might be straightforward if Alice and Bob are in the same place when they choose their secret key. But what if Alice and Bob are far apart, say on different continents? One possible solution is for Alice and Bob to use a public-key cryptosystem.

1.1.2 Public-key Cryptosystems

The revolutionary idea of ***public-key cryptography*** was introduced in the 1970s by Diffie and Hellman. Their idea was that it might be possible to devise a cryptosystem in which there are two distinct keys. A ***public key*** would be used to encrypt the plaintext and a ***private key*** would enable the ciphertext to be decrypted. Note that a public key can be known to "everyone," whereas a private key is known to only one person (namely, the recipient of the encrypted message). So a public-key cryptosystem would enable anyone to encrypt a message to be transmitted to Bob, say, and only Bob could decrypt the message. The first and best-known example of a public-key cryptosystem is the *RSA Cyptosystem* that

was invented by Rivest, Shamir and Adleman. Various types of public-key cryptosystems are presented in Chapters 6, 7, and 9.

Public-key cryptography obviates the need for two parties to agree on a prior shared secret key. However, it is still necessary to devise a method to distribute public keys securely. But this is not necessarily a trivial goal to accomplish, the main issue being the correctness or authenticity of purported public keys. Certificates, which we will discuss a bit later, are one common method to deal with this problem.

1.1.3 Block and Stream Ciphers

Cryptosystems are usually categorized as ***block ciphers*** or ***stream ciphers***. In a block cipher, the plaintext is divided into fixed-sized chunks called ***blocks***. A block is specified to be a bitstring (i.e., a string of 0's and 1's) of some fixed length (e.g., 64 or 128 bits). A block cipher will encrypt (or decrypt) one block at a time. In contrast, a stream cipher first uses the key to construct a ***keystream***, which is a bitstring that has exactly the same length as the plaintext (the plaintext is a bitstring of arbitrary length). The encryption operation constructs the ciphertext as the exclusive-or of the plaintext and the keystream. Decryption is accomplished by computing the exclusive-or of the ciphertext and the keystream. Public-key cryptosystems are invariably block ciphers, while secret-key cryptosystems can be block ciphers or stream ciphers. Block ciphers are studied in detail in Chapter 4.

1.1.4 Hybrid Cryptography

One of the drawbacks of public-key cryptosystems is that they are much slower than secret-key cryptosystems. As a consequence, public-key cryptosystems are mainly used to encrypt small amounts of data, e.g., a credit card number. However, there is a nice way to combine secret- and public-key cryptography to achieve the benefits of both. This technique is called ***hybrid cryptography***. Suppose that Alice wants to encrypt a "long" message and send it to Bob. Assume that Alice and Bob do not have a prior shared secret key. Alice can choose a random secret key and encrypt the plaintext, using a (fast) secret-key cryptosystem. Alice then encrypts this secret key using Bob's public key. Alice sends the ciphertext and the encrypted key to Bob. Bob first uses his private decryption key to decrypt the secret key, and then he uses this secret key to decrypt the ciphertext.

Notice that the "slow" public-key cryptosystem is only used to encrypt a short secret key. The much faster secret-key cryptosystem is used to encrypt the longer plaintext. Thus, hybrid cryptography (almost) achieves the efficiency of secret-key cryptography, but it can be used in a situation where Alice and Bob do not have a previously determined secret key.

1.2 Message Integrity

This section discusses various tools that help to achieve integrity of data, including message authentication codes (MACs), signature schemes, and hash functions.

Cryptosystems provide ***secrecy*** (equivalently, ***confidentiality***) against an eavesdropping adversary, which is often called a ***passive adversary***. A passive adversary is assumed to be able to access whatever information is being sent from Alice to Bob; see Figure 1.1. However, there are many other threats that we might want to protect against, particularly when an ***active adversary*** is present. An active adversary is one who can alter information that is transmitted from Alice to Bob.

Figure 1.2 depicts some of the possible actions of an active adversary. An active adversary might

- alter the information that is sent from Alice to Bob,
- send information to Bob in such a way that Bob thinks the information originated from Alice, or
- divert information sent from Alice to Bob in such a way that a third party (Charlie) receives this information instead of Bob.

Possible objectives of an active adversary could include fooling Bob (say) into accepting "bogus" information, or misleading Bob as to who sent the information to him in the first place.

We should note that encryption, by itself, cannot protect against these kinds of active attacks. For example, a stream cipher is susceptible to a ***bit-flipping attack***. If some ciphertext bits are "flipped" (i.e., 0's are replaced by 1's and vice versa), then the effect is to flip the corresponding plaintext bits. Thus, an adversary can modify the plaintext in a predictable way, even though the adversary does not know what the plaintext bits are.

There are various types of "integrity" guarantees that we might seek to provide, in order to protect against the possible actions of an active adversary. Such an adversary might change the information that is being transmitted from Alice to Bob (and note that this information may or may not be encrypted). Alternatively, the adversary might try to "forge" a message and send it to Bob, hoping that he will think that it originated from Alice. Cryptographic tools that protect against these and related types of threats can be constructed in both the secret-key and public-key settings. In the secret-key setting, we will briefly discuss the notion of a ***message authentication code*** (or ***MAC***). In the public-key setting, the tool that serves a roughly similar purpose is a ***signature scheme***.

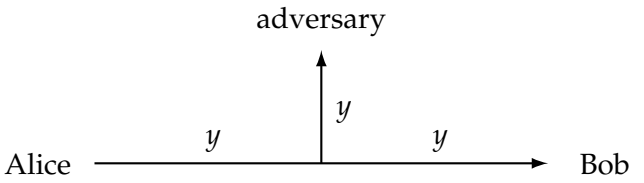

FIGURE 1.1: A passive adversary

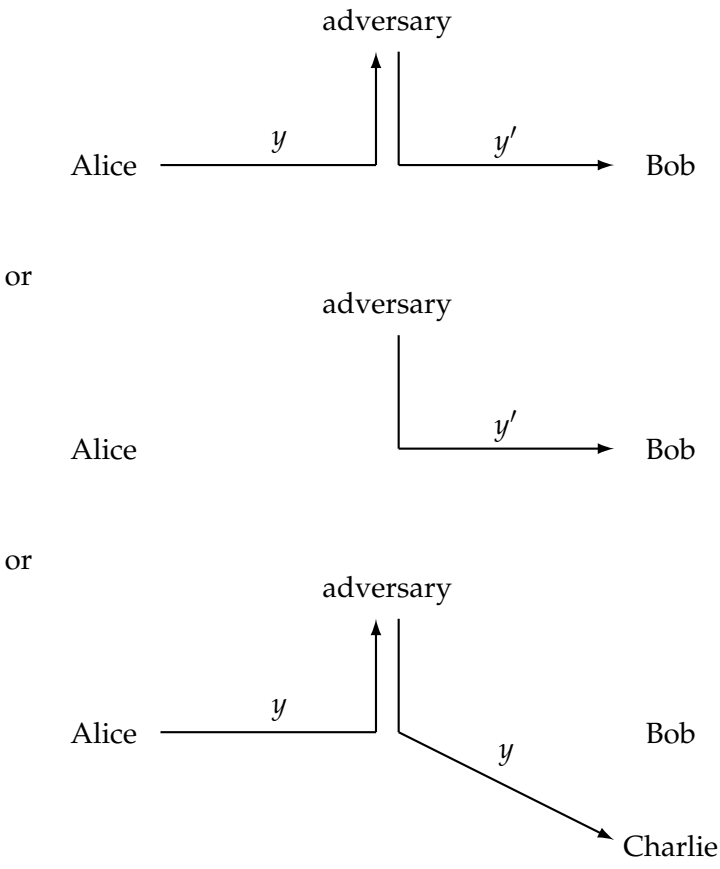

FIGURE 1.2: Active adversaries

1.2.1 Message Authentication Codes

A message authentication code requires Alice and Bob to share a secret key. When Alice wants to send a message to Bob, she uses the secret key to create a ***tag*** that she appends to the message (the tag depends on both the key and the message). When Bob receives the message and tag, he uses the key to re-compute the tag and checks to see if it is the same as the tag that he received. If so, Bob accepts the message as an authentic message from Alice; if not, then Bob rejects the message as being invalid. We note that the message may or may not be encrypted. MACs are discussed in Chapter 5.

If there is no need for confidentiality, then the message can be sent as plaintext. However, if confidentiality is desired, then the plaintext would be encrypted, and then the tag would be computed on the ciphertext. Bob would first verify the correctness of the tag. If the tag is correct, Bob would then decrypt the ciphertext. This process is often called ***encrypt-then-MAC*** (see Section 5.5.3 for a more detailed discussion of this topic).

For a MAC to be considered secure, it should be infeasible for the adversary to compute a correct tag for any message for which they have not already seen a valid tag. Suppose we assume that a secure MAC is being employed by Alice and Bob (and suppose that the adversary does not know the secret key that they are using). Then, if Bob receives a message and a valid tag, he can be confident that Alice created the tag on the given message (provided that Bob did not create it himself) and that neither the message nor the tag was altered by an adversary. A similar conclusion can be reached by Bob when he receives a message from Alice, along with a correct tag.

1.2.2 Signature Schemes

In the public-key setting, a signature scheme provides assurance similar to that provided by a MAC. In a signature scheme, the private key specifies a ***signing algorithm*** that Alice can use to sign messages. Similar to a MAC, the signing algorithm produces an output, which in this case is called a ***signature***, that depends on the message being signed as well as the key. The signature is then appended to the message. Notice that the signing algorithm is known only to Alice. On the other hand, there is a ***verification algorithm*** that is a public key (known to everyone). The verification algorithm takes as input a message and a signature, and outputs *true* or *false* to indicate whether the signature should be accepted as valid. One nice feature of a signature scheme is that anyone can verify Alice's signatures on messages, provided that they have an authentic copy of Alice's verification key. In contrast, in the MAC setting, only Bob can verify tags created by Alice (when Alice and Bob share a secret key). Signature schemes are studied in Chapter 8.

Security requirements for signature schemes are similar to MACs. It should be infeasible for an adversary to create a valid signature on any message not previously signed by Alice. Therefore, if Bob (or anyone else) receives a message and a valid tag (i.e., one that can be verified using Alice's public verification algorithm),

then the recipient can be confident that the signature was created by Alice and neither the message nor the signature was modified by an adversary.

One common application of signatures is to facilitate secure software updates. When a user purchases software from an online website, it typically includes a verification algorithm for a signature scheme. Later, when an updated version of the software is downloaded, it includes a signature (on the updated software). This signature can be verified using the verification algorithm that was downloaded when the original version of the software was purchased. This enables the user's computer to verify that the update comes from the same source as the original version of the software.

Signature schemes can be combined with public-key encryption schemes to provide confidentiality along with the integrity guarantees of a signature scheme. Assume that Alice wants to send a signed, encrypted (short) message to Bob. In this situation, the most commonly used technique is for Alice to first create a signature on the plaintext using her private signing algorithm, and then encrypt the plaintext and signature using Bob's public encryption key. When Bob receives the message, he first decrypts it, and then he checks the validity of the signature. This process is called ***sign-then-encrypt***; note that this is in some sense the reverse of the "encrypt-then-MAC" procedure that is used in the secret-key setting.

1.2.3 Nonrepudiation

There is one somewhat subtle difference between MACs and signature schemes. In a signature scheme, the verification algorithm is public. This means that the signature can be verified by anyone. So, if Bob receives a message from Alice containing her valid signature on the message, he can show the message and the signature to anyone else and be confident that the third party will also accept the signature as being valid. Consequently, Alice cannot sign a message and later try to claim that she did not sign the message, a property that is termed ***nonrepudiation***. This is useful in the setting of contracts, where we do not want someone to be able to renege on a signed contract by claiming (falsely) that their signature has been "forged," for example.

However, for a MAC, there is no third-party verifiability because the secret key is required to verify the correctness of the tag, and the key is known only to Alice and Bob. Even if the secret key is revealed to a third party (e.g., as a result of a court order), there is no way to determine if the tag was created by Alice or by Bob, because anything Bob can do, Alice can do as well, and vice versa. So a MAC does not provide nonrepudiation, and for this reason, a MAC is sometimes termed "deniable." It is interesting to note, however, that there are situations where deniability is desirable. This could be the case in real-time communications, where Alice and Bob want to be assured of the authenticity of their communications as they take place, but they do not want a permanent, verifiable record of this communication to exist. Such communication is analogous to an "off-the-record" conversation, e.g., between a journalist and an anonymous source. A MAC is useful in the con-

text of conversations of this type, especially if care is taken, after the conversation is over, to delete the secret keys that are used during the communication.

1.2.4 Certificates

We mentioned that verifying the authenticity of public keys, before they are used, is important. A certificate is a common tool to help achieve this objective. A ***certificate*** will contain information about a particular user or, more commonly, a website, including the website's public keys. These public keys will be signed by a trusted authority. It is assumed that everyone has possession of the trusted authority's public verification key, so anyone can verify the trusted authority's signature on a certificate. See Section 8.6 for more information about certificates.

This technique is used on the internet in *Transport Layer Security* (which is commonly called *TLS*). When a user connects to a secure website, say one belonging to a business engaged in electronic commerce, the website of the company will send a certificate to the user so the user can verify the authenticity of the website's public keys. These public keys will subsequently be used to set up a secure channel, between the user and the website, in which all information is encrypted. Note that the public key of the trusted authority, which is used to verify the public key of the website, is typically hard-coded into the web browser.

1.2.5 Hash Functions

Signature schemes tend to be much less efficient than MACs. So it is not advisable to use a signature scheme to sign "long" messages. (Actually, most signature schemes are designed to only sign messages of a short, fixed length.) In practice, messages are "hashed" before they are signed. A ***cryptographic hash function*** is used to compress a message of arbitrary length to a short, random-looking, fixed-length ***message digest***. Note that a hash function is a public function that is assumed to be known to everyone. Further, a hash function has no key. Hash functions are discussed in Chapter 5.

After Alice hashes the message, she signs the message digest, using her private signing algorithm. The original message, along with the signature on the message, is then transmitted to Bob, say. This process is called ***hash-then-sign***. To verify the signature, Bob will compute the message digest by hashing the message. Then he will use the public verification algorithm to check the validity of the signature on the message digest. When a signature is used along with public-key encryption, the process would actually be ***hash-then-sign-then-encrypt***. That is, the message is hashed, the message digest is then signed, and finally, the message and signature are encrypted.

A cryptographic hash function is very different from a hash function that is used to construct a hash table, for instance. In the context of hash tables, a hash function is generally required only to yield collisions[1] with a sufficiently small probability. On the other hand, if a cryptographic hash function is used, it should

[1] A ***collision*** for a function h occurs when $h(x) = h(y)$ for some $x \neq y$.

be computationally infeasible to find collisions, even though they must exist. Cryptographic hash functions are usually required to satisfy additional security properties, as discussed in Section 5.2.

Cryptographic hash functions also have other uses, such as for ***key derivation***. When used for key derivation, a hash function would be applied to a long random string in order to create a short random key.

Finally, it should be emphasized that hash functions cannot be used for encryption, for two fundamental reasons. First is the fact that hash functions do not have a key. The second is that hash functions cannot be inverted (they are not injective functions) so a message digest cannot be "decrypted" to yield a unique plaintext value.

1.3 Cryptographic Protocols

Cryptographic tools such as cryptosystems, signature schemes, hash functions, etc., can be used on their own to achieve specific security objectives. However, these tools are also used as components in more complicated protocols. (Of course, protocols can also be designed "from scratch," without making use of prior primitives.)

In general, a ***protocol*** (or ***interactive protocol***) refers to a specified sequence of messages exchanged between two (or possibly more) parties. A ***session*** of a protocol between Alice and Bob, say, will consist of one or more ***flows***, where each flow consists of a message sent from Alice to Bob or vice versa. At the end of the session, the parties involved may have established some common shared information, or confirmed possession of some previously shared information.

One important type protocol is an ***identification scheme***, in which one party "proves" their identity to another by demonstrating possession of a password, for example. More sophisticated identification protocols will instead consist of two (or more) flows, for example a challenge followed by a response, where the response is computed from the challenge using a certain secret or private key. Identification schemes are the topic of Chapter 10.

There are many kinds of protocols associated with various aspects of choosing keys or communicating keys from one party to another. In a ***key distribution scheme***, keys might be chosen by a trusted authority and communicated to one or more members of a certain network. Another approach, which does not require the participation of an active trusted authority, is called ***key agreement***. In a key agreement scheme, Alice and Bob (say) are able to end up with a common shared secret key, which should not become known to an adversary. These and related topics are discussed in Chapters 11 and 12.

A ***secret sharing scheme*** involves a trusted authority distributing "pieces" of information (called "shares") in such a way that certain subsets of shares can be suitably combined to reconstruct a certain predefined secret. One common type

of secret sharing scheme is a ***threshold scheme***. In a (k, n)-threshold scheme, there are n shares, and any k shares permit the reconstruction of the secret. On the other hand, $k-1$ or fewer shares provide no information about the value of the secret. Secret sharing schemes are studied in Chapter 11.

1.4 Security

A fundamental goal for a cryptosystem, signature scheme, etc., is for it to be "secure." But what does it mean to be secure and how can we gain confidence that something is indeed secure? Roughly speaking, we would want to say that an adversary cannot succeed in "breaking" a cryptosystem, for example, but we have to make this notion precise. Security in cryptography involves consideration of three different aspects: an ***attack model***, an ***adversarial goal***, and a ***security level***. We will discuss each of these in turn.

The attack model specifies the information that is available to the adversary. We will always assume that the adversary knows the scheme or protocol being used (this is called ***Kerckhoffs' Principle***). The adversary is also assumed to know the public key (if the system is a public-key system). On the other hand, the adversary is assumed not to know any secret or private keys being used. Possible additional information provided to the adversary should be specified in the attack model.

The adversarial goal specifies exactly what it means to "break" the cryptosystem. What is the adversary attempting to do and what information are they trying to determine? Thus, the adversarial goal defines a "successful attack."

The security level attempts to quantify the effort required to break the cryptosystem. Equivalently, what computational resources does the adversary have access to and how much time would it take to carry out an attack using those resources?

A statement of security for a cryptographic scheme will assert that a particular adversarial goal cannot be achieved in a specified attack model, given specified computational resources.

We now illustrate some of the above concepts in relation to a cryptosystem. There are four commonly considered attack models. In a ***known ciphertext attack***, the adversary has access to some amount of ciphertext that is all encrypted with the same unknown key. In a ***known plaintext attack***, the adversary gains access to some plaintext as well as the corresponding ciphertext (all of which is encrypted with the same key). In a ***chosen plaintext attack***, the adversary is allowed to choose plaintext, and then they are given the corresponding ciphertext. Finally, in a ***chosen ciphertext attack***, the adversary chooses some ciphertext and they are then given the corresponding plaintext.

Clearly a chosen plaintext or chosen ciphertext attack provides the adversary with more information than a known ciphertext attack. So they would be con-

sidered to be stronger attack models than a known ciphertext attack, since they potentially make the adversary's job easier.

The next aspect to study is the adversarial goal. In a ***complete break*** of a cryptosystem, the adversary determines the private (or secret) key. However, there are other, weaker goals that the adversary could potentially achieve, even if a complete break is not possible. For example, the adversary might be able to decrypt a previously unseen ciphertext with some specified non-zero probability, even though they have not been able to determine the key. Or, the adversary might be able to determine some partial information about the plaintext, given a previously unseen ciphertext, with some specified non-zero probability. "Partial information" could include the values of certain plaintext bits. Finally, as an example of a weak goal, the adversary might be able distinguish between encryptions of two given plaintexts.[2]

Other cryptographic primitives will have different attack models and adversarial goals. In a signature scheme, the attack model would specify what kind of (valid) signatures the adversary has access to. Perhaps the adversary just sees some previously signed messages, or maybe the adversary can request the signer to sign some specific messages of the adversary's choosing. The adversarial goal is typically to sign some "new" message (i.e., one for which the adversary does not already know a valid signature). Perhaps the adversary can find a valid signature for some specific message that the adversary chooses, or perhaps they can find a valid signature for any message. These would represent weak and strong adversarial goals, respectively.

Three levels of security are often studied, which are known as ***computational security***, ***provable security***, and ***unconditional security***.

Computational security means that a specific algorithm to break the system is computationally infeasible, i.e., it cannot be accomplished in a reasonable amount of time using currently available computational resources. Of course, a system that is computationally secure today may not be computationally secure indefinitely. For example, new algorithms might be discovered, computers may get faster, or fundamental new computing paradigms such as quantum computing might become practical. Quantum computing, if it becomes practical, could have an enormous impact on the security of many kinds of public-key cryptography; this is addressed in more detail in Section 9.1.

It is in fact very difficult to predict how long something that is considered secure today will remain secure. There are many examples where many cryptographic schemes have not survived as long as originally expected due to the reasons mentioned above. This has led to rather frequent occurrences of replacing standards with improved standards. For example, in the case of hash functions, there have been a succession of proposed and/or approved standards, denoted as *SHA-0*, *SHA-1*, *SHA-2* and *SHA-3*, as new attacks have been found and old standards have become insecure.

[2] Whether or not this kind of limited information can be exploited by the adversary in a malicious way is another question, of course.

An interesting example relating to broken predictions is provided by the public-key *RSA Cryptosystem*. In the August 1977 issue of *Scientific American*, the eminent mathematical expositor Martin Gardner wrote a column on the newly developed *RSA* public-key cryptosystem entitled "A new kind of cipher that would take millions of years to break." Included in the article was a challenge ciphertext, encrypted using a 512-bit key. However, the challenge was solved 17 years later, on April 26, 1994, by factoring the given public key (the plaintext was "the magic words are squeamish ossifrage"). The statement that the cipher would take millions of years to break probably referred to how long it would take to run the best factoring algorithm known in 1977 on the fastest computer available in 1977. However, between 1977 and 1994, there were several developments, including the following:

- computers became much faster,
- improved factoring algorithms were found, and
- the development of the internet facilitated large-scale distributed computations.

Of course, it is basically impossible to predict when new algorithms will be discovered. Also, the third item listed above can be regarded as a "paradigm shift" that was probably not on anyone's radar in 1977.

The next "level" of security we address is provable security (also known as ***reductionist security***), which refers to a situation where breaking the cryptosystem (i.e., achieving the adversarial goal) can be reduced in a complexity-theoretic sense to solving some underlying (assumed difficult) mathematical problem. This would show that breaking the cryptosystem is at least as difficult as solving the given hard problem. Provable security often involves reductions to the factoring problem or the discrete logarithm problem (these problems are studied in Sections 6.6 and 7.2, respectively).

Finally, unconditional security means that the cryptosystem cannot be broken (i.e., the adversarial goal is not achievable), even with unlimited computational resources, because there is not enough information available to the adversary (as specified in the attack model) for them to be able to do this. The most famous example of an unconditionally secure cryptosystem is the *One-time Pad*. In this cryptosystem, the key is a random bitstring having the same length as the plaintext. The ciphertext is formed as the exclusive-or of the plaintext and the key. For the *One-time Pad*, it can be proven mathematically that the adversary can obtain no partial information whatsoever about the plaintext (other than its length), given the ciphertext, provided the key is used to encrypt only one string of plaintext and the key has the same length as the plaintext. The *One-time Pad* is discussed in Chapter 3.

When we analyze a cryptographic scheme, our goal would be to show that the adversary cannot achieve a *weak* adversarial goal in a *strong* attack model, given *significant* computational resources.

The preceding discussion of security has dealt mostly with the situation of a cryptographic primitive such as a cryptosystem. However, cryptographic primitives are generally combined in complicated ways when protocols are defined and ultimately implemented. Even seemingly simple implementation decisions can lead to unexpected vulnerabilities. For example, when data is encrypted using a block cipher, it first needs to be split into fixed length chunks, e.g., 128-bit blocks. If the data does not exactly fill up an integral number of blocks, then some padding has to be introduced. It turns out that a standard padding technique, when used with the common CBC mode of operation, is susceptible to an attack known as a ***padding oracle attack***, which was discovered by Vaudenay in 2002 (see Section 4.7.1 for a description of this attack).

There are also various kinds of attacks against physical implementations of cryptography that are known as ***side channel attacks***. Examples of these include ***timing attacks***, ***fault analysis attacks***, ***power analysis attacks***, and ***cache attacks***. The idea is that information about a secret or private key might be leaked by observing or physically manipulating a device (such as a smart card) on which a particular cryptographic scheme is implemented. One example would be observing the time taken by the device to perform certain computations (a so-called "timing attack"). This leakage of information can take place even though the scheme is "secure."

1.5 Notes and References

There are many monographs and textbooks on the subject of cryptography. We will mention here a few general treatments that may be useful to readers.

For an accessible, non-mathematical treatment, we recommend

- *Everyday Cryptography: Fundamental Principles and Applications, Second Edition* by Keith Martin [127].

For a more mathematical point of view, the following recent texts are helpful:

- *An Introduction to Mathematical Cryptography* by J. Hoffstein, J. Pipher, and J. Silverman [96]
- *Introduction to Modern Cryptography, Second Edition* by J. Katz and Y. Lindell [104]
- *Understanding Cryptography: A Textbook for Students and Practitioners* by C. Paar and J. Pelzl [157]
- *Cryptography Made Simple* by Nigel Smart [185]
- *A Classical Introduction to Cryptography: Applications for Communications Security* by Serge Vaudenay [196].

For mathematical background, especially for public-key cryptography, we recommend

- *Mathematics of Public Key Cryptography* by Stephen Galbraith [84].

Finally, the following is a valuable reference, even though it is quite out of date:

- *Handbook of Applied Cryptography* by A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone [134].

References

Carlisle Adams and Steve Lloyd . Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition. Addison Wesley, 2003.

Leonard Adleman . A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In 20th Annual Symposium on Foundations of Computer Science, pages 55–60. IEEE, 1979.

Werner Alexi , Benny Chor , Oded Goldreich , and Claus Schnorr . RSA and Rabin functions: certain parts are as hard as the whole. SIAM Journal on Computing, 17 (1988), 194–209.

Jee Hea An , Yevgeniy Dodis , and Tal Rabin . On the security of joint signature and encryption. Lecture Notes in Computer Science, 2332 (2002), 83–107. (EUROCRYPT 2002.)

Razvan Barbulescu , Pierrick Gaudry , Antoine Joux , and Emmanuel ThomÉ . A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. Lecture Notes in Computer Science, 8441 (2014), 1–16. (EUROCRYPT 2014.)

Elaine Barker and John Kelsey . Recommendation for random number generation using deterministic random bit generators. National Institute of Standards and Technology (NIST) Special Publication 800-90A, 2012.

Elaine Barker and John Kelsey . Recommendation for random number generation using deterministic random bit generators. National Institute of Standards and Technology (NIST) Special Publication 800-90A, Revision 1, 2015.

Elaine Barker and Allen Roginsky . Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths. National Institute of Standards and Technology (NIST) Special Publication 800-131A, revision 1, 2015.

Elaine Barker , Lily Chen , and Rich Davis . Recommendation for Key-Derivation Methods in Key-Establishment Schemes. Draft National Institute of Standards and Technology (NIST) Special Publication 800-56C, revision 1, August 2017.

Friedrich Bauer . Decrypted Secrets: Methods and Maxims of Cryptology, Second Edition. Springer, 2000.

Pierre Beauchemin and Gilles Brassard . A generalization of Hellman’s extension to Shannon’s approach to cryptography. Journal of Cryptology, 1 (1988), 129–131.

Pierre Beauchemin , Gilles Brassard , Claude CrEpeau Claude Goutier, and Carl Pomerance. The generation of random numbers that are probably prime. Journal of Cryptology, 1 (1988), 53–64.

Henry Beker and Fred Piper . Cipher Systems: The Protection of Communications. John Wiley and Sons, 1983.

Mihir Bellare , Shafi Goldwasser , and Daniele Micciancio . “Pseudo-random” number generation within cryptographic algorithms: the DSS case. Lecture Notes in Computer Science, 1294 (1997), 277–292. (CRYPTO ’97.)

Mihir Bellare , Joe Kilian , and Phillip Rogaway . The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences, 61 (2000), 362–399.

Mihir Bellare and Chanathip Namprempre . Authenticated encryption: relations among notions and analysis of the generic composition paradigm. Lecture Notes in Computer Science 1976, (2000), 531-545. (ASI-ACRYPT 2000.)

Mihir Bellare and Adrian Palacio . GQ and Schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks. Lecture Notes in Computer Science, 2442 (2002), 162–177. (CRYPTO 2002.)

Mihir Bellare and Phillip Rogaway . Entity authentication and key distribution. Lecture Notes in Computer Science, 773 (1994), 232–249. (CRYPTO ’93.)

Mihir Bellare and Phillip Rogaway . Optimal asymmetric encryption. Lecture Notes in Computer Science, 950 (1995), 92–111. (EUROCRYPT ’94.)

Mihir Bellare and Phillip Rogaway . Provably secure session key distribution: the three party case. In 27th Annual ACM Symposium on Theory of Computing, pages 57–66. ACM Press, 1995.

Mihir Bellare and Phillip Rogaway . The exact security of digital signatures: how to sign with RSA and Rabin. Lecture Notes in Computer Science, 1070 (1996), 399–416. (EUROCRYPT ’96.)

Mihir Bellare and Phillip Rogaway . Random oracles are practical: a paradigm for designing efficient protocols. In First ACM Conference on Computer and Communications Security, pages 62–73. ACM Press, 1993.

Daniel Bernstein , Tanja Lange , and Christiane Peters . Attacking and defending the McEliece cryptosystem. Lecture Notes in Computer Science, 5299 (2008), 31–46. (PQCrypto 2008.)

Daniel Bernstein , Johannes Buchmann , and Erik Dahmen , Eds. Post-quantum Cryptography. Springer, 2009.

Daniel Bernstein and Tanja Lange . Post-quantum cryptography. Nature 549 (2017), 188–194.

Guido Bertoni , Joan Daemen , MichaËl Peeters , and Gilles Van Assche . Sponge functions. Ecrypt Hash Workshop 2007, May 2007. Available from https://keccak.team/files/SpongeFunctions.pdf.

Guido Bertoni , Joan Daemen , MichaËl Peeters , and Gilles Van Assche . The Keccak SHA-3 submission, January 2011. Available from https://keccak.team/files/Keccak-submission-3.pdf.

Albrecht Beutelspacher . Cryptology. Mathematical Association of America, 1994.

Eli Biham and Adi Shamir . Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4 (1991), 3–72.

Alex Biryukov , Daniel Dinu , and Dmitry Khovratovich . Argon2: new generation of memory-hard functions for password hashing and other applications. In IEEE European Symposium on Security and Privacy, pages 292–302. IEEE, 2016.

Alex Biryukov , Orr Dunkelman , Nathan Keller , Dmitry Khovratovich , and Adi Shamir . Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds. Lecture Notes in Computer Science, 6110 (2010), 299–319. (EUROCRYPT 2010.)

John Black , Shai Halevi , Hugo Krawczyk , Ted Krovetz , and Phillip Rogaway . UMAC: fast and secure message authentication. Lecture Notes in Computer Science, 1666 (1999), 216–233. (CRYPTO ’99.)

Simon Blake-Wilson and Alfred Menezes . Entity authentication and authenticated key transport protocols employing asymmetric techniques. Lecture Notes in Computer Science, 1361 (1998), 137–158. (Fifth International Workshop on Security Protocols.)

Simon Blake-Wilson and Alfred Menezes , Authenticated Diffie-Hellman key agreement protocols. Lecture Notes in Computer Science, 1556 (1999), 339–361. (Selected Areas in Cryptography ’98.)

G. R. (Bob) Blakley . Safeguarding cryptographic keys. Federal Information Processing Standard Conference Proceedings, 48 (1979), 313–317.

R. Blom . An optimal class of symmetric key generation schemes. Lecture Notes in Computer Science, 209 (1985), 335–338. (EUROCRYPT ’84.)

Lenore Blum , Manuel Blum , and Michael Shub . A simple unpredictable random number generator. SIAM Journal on Computing, 15 (1986), 364–383.

Carlo Blundo , Alfredo De Santis , Amir Herzberg , Shay Kutten , Ugo Vaccaro , and Moti Yung . Perfectly-secure key distribution for dynamic conferences. Lecture Notes in Computer Science, 740 (1993), 471–486. (CRYPTO ’92.)
Andrey Bogdanov , Dmitry Khovratovich , and Christian Rech-berger . Biclique cryptanalysis of the full AES. Lecture Notes in Computer Science, 7073 (2011), 344–371. (ASIACRYPT 2011.)
Dan Boneh . The decision Diffie-Hellman problem. Lecture Notes in Computer Science, 1423 (1998), 48–63. (Proceedings of the Third Algorithmic Number Theory Symposium.)
Dan Boneh and Glenn Durfee . Cryptanalysis of RSA with private key d less than N0.292 . IEEE Transactions on Information Theory, 46 (2000), 1339–1349.
Dan Boneh and Matthew Franklin . Identity-based encryption from the Weil pairing. Lecture Notes in Computer Science, 2139 (2001), 213–229. (CRYPTO 2001.)
Dan Boneh and James Shaw . Collusion-secure fingerprinting for digital data. IEEE Transactions on Information Theory, 44 (1998), 1897–1905.
Colin Boyd and Anish Mathuria . Protocols for Authentication and Key Establishment. Springer, 2003.
Gilles Brassard and Paul Bratley . Fundamentals of Algorithmics. Prentice Hall, 1995.
Richard Brent . An improved Monte Carlo factorization method. BIT, 20 (1980), 176–184.
David Bressoud and Stan Wagon . A Course in Computational Number Theory. Wiley, 2008.
Jon Brodkin . Kim Dotcom claims he invented two-factor authentication but he wasn’t first. Ars Technica, May 23, 2013. https://arstechnica.com/information-technology/2013/05/kim-dotcom-claims-he-invented-two-factor-authentication-but-he-wasnt-first/
Daniel R. L. Brown and Kristian Gjøsteen . A security analysis of the NIST SP 800-90 elliptic curve random number generator. Lecture Notes in Computer Science, 4622 (2007), 466–481. (CRYPTO 2007.)
Johannes Buchmann , Erik Dahmen , and Andreas Hülsing . XMSS — a practical forward secure signature scheme based on minimal security assumptions. Lecture Notes in Computer Science, 7071 (2011), 117–129. (PQCrypto 2011.)
Mike Burmester . On the risk of opening distributed keys. Lecture Notes in Computer Science, 839 (1994), 308–317 (CRYPTO ’94.)
Mike Burmester and Yvo Desmedt . A secure and efficient conference key distribution system. Lecture Notes in Computer Science, 950 (1994), 275–286 (EUROCRYPT ’94.)
David Burton . Elementary Number Theory, 7th Edition. McGraw-Hill, 2010.
R. Canetti and h. Krawczyk . Analysis of key-exchange protocols and their use for building secure channels. Lecture Notes in Computer Science, 2045 (2001), 453–474 (EUROCRYPT 2001.)
J. Lawrence Carter and Mark Wegman . Universal classes of hash functions. Journal of Computer and System Sciences, 18 (1979), 143–154.
Mark Chateauneuf , Alan Ling , and Douglas Stinson . Slope packings and coverings, and generic algorithms for the discrete logarithm problem. Journal of Combinatorial Designs, 11 (2003), 36–50.
Benny Chor , Amos Fiat , Moni Naor , and Benny Pinkas . Tracing traitors. IEEE Transactions on Information Theory, 46 (2000), 893–910.
Carlos Cid , Sean Murphy , and Matthew Robshaw . Algebraic Aspects of the Advanced Encryption Standard. Springer, 2006.
Clifford Cocks . An identity based encryption scheme based on quadratic residues. Lecture Notes in Computer Science, 2260 (2001), 360–363. (Eighth IMA International Conference on Cryptography and Coding.)
Katriel Cohn-Gordon , Cas Cremers , Benjamin Dowling , Luke Garratt , and Douglas Stebila . A formal security analysis of the Signal messaging protocol. Cryptology ePrint Archive: Report 2016/1013. https://eprint.iacr.org/2016/1013.pdf
Don Coppersmith . Fast evaluation of logarithms in fields of characteristic two IEEE Transactions on Information Theory 30 (1984), 587–594.
Nicolas T. Courtois . Fast algebraic attacks on stream ciphers with linear feedback. Lecture Notes in Computer Science, 2729 (2003), 176–194. (CRYPTO 2003.)
Joan Daemen and Vincent Rijmen . The Design of Rijndael: AES — The Advanced Encryption Standard. Springer, 2002.
Ivan Damgaard . A design principle for hash functions. Lecture Notes in Computer Science, 435 (1990), 416–427. (CRYPTO ’89.)
Christophe De CanniEre . Trivium: a stream cipher construction inspired by block cipher design principles. Lecture Notes in Computer Science, 4176 (2006), 171–186. (International Conference on Information Security, ISC 2006.)
Christophe De CanniEre and Bart Preneel . Trivium: a stream cipher construction inspired by block cipher design principles. eSTREAM submitted papers, available from http://www.ecrypt.eu.org/stream/papersdir/2006/021.pdf.
John DeLaurentis . A further weakness in the common modulus protocol for the RSA cryptosystem. Cryptologia, 8 (1984), 253–259.
Dorothy Denning and Giovanni Sacco . Timestamps in key distribution protocols. Communications of the ACM, 24 (1981), 533–536.
Whitfield Diffie . The first ten years of public-key cryptography. In Contemporary Cryptology, The Science of Information Integrity, pages 135–175. IEEE Press, 1992.
Whitfield Diffie and Martin Hellman . Multiuser cryptographic techniques. Federal Information Processing Standard Conference Proceedings, 45 (1976), 109–112.
Whitfield Diffie and Martin Hellman . New directions in cryptography. IEEE Transactions on Information Theory, 22 (1976), 644–654.
Whitfield Diffie , Paul Van Oorschot , and Michael Wiener . Authentication and authenticated key exchanges. Designs, Codes and Cryptography, 2 (1992), 107–125.
Jintai Ding and Dieter Schmidt . Rainbow, a new multivariable polynomial signature scheme. Lecture Notes in Computer Science, 3531 (2005), 164–175. (ACNS 2005.)
Jintai Ding , Xiang Xie , and Xiaodong Lin . A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive: Report 2012/688. https://eprint.iacr.org/2012/688.pdf
Chris Dods , Nigel Smart , and Martijn Stam . Hash based digital signature schemes. Lecture Notes in Computer Science, 3796 (2006), 96–116. (Cryptography and Coding 2005.)
Morris Dworkin . Recommendation for Block Cipher Modes of Operation: Methods and Techniques. National Institute of Standards and Technology (NIST) Special Publication 800-38A, 2001.
Morris Dworkin . Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication. National Institute of Standards and Technology (NIST) Special Publication 800-38B, 2005 (updated 2016).
Morris Dworkin . Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality. National Institute of Standards and Technology (NIST) Special Publication 800-38D, 2004.

Morris Dworkin . Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. National Institute of Standards and Technology (NIST) Special Publication 800-38D, 2007.
Taher ElGamal . A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31 (1985), 469–472.
Andreas Enge . Bilinear pairings on elliptic curves. ArXiv report 1301.5520v2, Feb. 15, 2014. https://arxiv.org/abs/1301.5520v2.
Uriel Feige , Amos Fiat , and Adi Shamir . Zero-knowledge proofs of identity. Journal of Cryptology, 1 (1988), 77–94.
Amos Fiat and Adi Shamir . How to prove yourself: practical solutions to identification and signature problems. Lecture Notes in Computer Science, 263 (1987), 186–194. (CRYPTO '86.)
Stephen Galbraith . Mathematics of Public Key Cryptography. Cambridge University Press, 2012.
Stephen Galbraith and Pierrick Gaudry . Recent progress on the elliptic curve discrete logarithm problem. Designs, Codes and Cryptography, 78 (2016), 51–72.
Craig Gentry . Fully homomorphic encryption using ideal lattices. In 41st Annual Symposium on Theory of Computing, pages 169–178. ACM, 2009.
Edgar N. Gilbert , F. Jessie MacWilliams , and Neil J. A. Sloane . Codes which detect deception. Bell Systems Technical Journal, 53 (1974), 405–424.
Charles Goldie and Richard Pinch . Communication Theory. Cambridge University Press, 1991.
Shafi Goldwasser and Silvio Micali . Probabilistic encryption. Journal of Computer and Systems Science, 28 (1984), 270–299.
Shafi Goldwasser , Silvio Micali , and Po Tong . Why and how to establish a common code on a public network. In 23rd Annual Symposium on the Foundations of Computer Science, pages 134–144. IEEE Press, 1982.
Thomas C. Hales . The NSA Back Door to NIST. Notices of the AMS, 61 (2014), 190–192.
Darrel Hankerson , Alfred Menezes , and Scott Vanstone . Guide to Elliptic Curve Cryptography. Springer, 2004.
Howard M. Heys . A tutorial on linear and differential cryptanalysis. Cryptologia, 26 (2002), 189–221.
Howard M. Heys and Stafford E. Tavares . Substitution-permutation networks resistant to differential and linear cryptanalysis. Journal of Cryptology, 9 (1996), 1–19.
M. Jason Hinek . Cryptanalysis of RSA and Its Variants. Chapman and Hall/CRC, 2009.
Jeffrey Hoffstein , Jill Pipher , and Joseph Silverman . An Introduction to Mathematical Cryptography. Springer, 2008.
Henk Hollmann , Jack van Lint , Jean-Paul Linnartz , and Ludo Tolhuizen . On codes with the identifiable parent property. Journal of Combinatorial Theory A, 82 (1998), 121–133.
W. Cary Huffman and Vera Pless . Fundamentals of Error-Correcting Codes. Cambridge University Press, 2003.
Tetsu Iwata and Kaoru Kurosawa . OMAC: one-key CBC MAC. Lecture Notes in Computer Science, 2887 (2003), 129–153. (Fast Software Encryption 2003.)
Don Johnson , Alfred Menezes , and Scott Vanstone . The elliptic curve digital signature algorithm (ECDSA). International Journal on Information Security, 1 (2001), 36–63.
Antoine Joux . A new index calculus algorithm with complexity $L(1/4 + o(1))$ in small characteristic. Lecture Notes in Computer Science, 8282 (2014), 355–379. (Selected Areas in Cryptography 2013.)
Antoine Joux , Andrew Odlyzko , and Cecile Pierro . The past, evolving present, and future of the discrete logarithm. In Open Problems in Mathematics and Computational Science, pages 5–36. Springer, 2014
David Kahn . The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet. Scribner, 1996.
Jonathan Katz and Yehuda Lindell . Introduction to Modern Cryptography, Second Edition. Chapman and Hall/CRC, 2014.
Aviad Kipnis , Jacques Patarin , and Louis Goubin . Unbalanced oil and vinegar signature schemes. Lecture Notes in Computer Science, 1592 (1999), 206–222. (EUROCRYPT '99.)
Rudolf Kippenhahn . Code Breaking, A History and Exploration. Overlook Press, 1999.
Andreas Klein . Stream Ciphers. Springer, 2013.
Thorsten Kleinjung , Kazumaro Aoki , Jens Franke , Arjen Lenstra , Emmanuel ThomÉ , Joppe Bos , Pierrick Gaudry , Alexander Kruppa , Peter Montgomery , Dag Arne Osvik , Herman te Riele , Andrey Timofeev , and Paul Zimmermann . Factorization of a 768-Bit RSA modulus. Lecture Notes in Computer Science, 6223 (2010), 333–350. (CRYPTO 2010.)
Lars R. Knudsen and Matthew Robshaw . The Block Cipher Companion. Springer, 2011.
Donald E. Knuth . The Art of Computer Programming, Volume 2, Seminumerical Algorithms, Second Edition. Addison-Wesley, 1998.
Neal Koblitz . A Course in Number Theory and Cryptography, Second Edition. Springer, 1994.
Neal Koblitz . Elliptic curve cryptosystems. Mathematics of Computation, 48 (1987), 203–209.
Neal Koblitz and Alfred Menezes . Another look at HMAC. Journal of Mathematical Cryptology 7 (2013), 225–251.
John T. Kohl and B. Clifford Neuman . The Kerberos Network Authentication Service (V5). Network Working Group Request for Comments 1510, 1993.
John T. Kohl , B. Clifford Neuman , and Theodore Y. T'so . The evolution of the Kerberos authentication system. In Distributed Open Systems pages 78–94. IEEE Computer Society Press, 1994.
Loren M. Kohnfelder . Towards a practical public-key cryptosystem. Bachelor's Thesis, MIT, 1978.
Kaoru Kurosawa , Toshiya Ito , and Masaschi Takeuchi . Public key cryptosystem using a reciprocal number with the same intractability as factoring a large number. Cryptologia, 12 (1988), 225–233.
David Lay , Steven Lay , and Judi McDonald . Linear Algebra and Its Applications, 5th Edition. Pearson, 2015.
Jooyoung Lee and Douglas R. Stinson . A combinatorial approach to key predistribution for distributed sensor networks. IEEE Wireless Communications and Networking Conference (WCNC 2005), vol. 2, pp. 1200-1205.
Jooyoung Lee and Douglas R. Stinson . On the construction of practical key predistribution schemes for distributed sensor networks using combinatorial designs. ACM Transactions on Information and System Security 11-2 (2008), article No. 1,35 pp.
Arjen Lenstra and Hendrik Lenstra, Jr. (Eds.) The Development of the Number Field Sieve. Lecture Notes in Mathematics, vol. 1554. Springer, 1993.
Rudolf Lidl and Harald Niederreiter . Finite Fields, Second Edition. Cambridge University Press, 1997.
Jan C. A. van der Lubbe . Basic Methods of Cryptography. Cambridge, 1998.

Michael Luby . Pseudorandomness and Cryptographic Applications. Princeton University Press, 1996.
F. Jessie MacWilliams and Neil J. A. Sloane . The Theory of Error-correcting Codes, North-Holland, 1977.
Moxie Marlinspike and Trevor Perrin (editor). The X3DH key agreement protocol. Open Whisper Systems, November 4, 2016. https://signal.org/docs/specifications/x3dh/
Keith M. Martin . Everyday Cryptography: Fundamental Principles and Applications, Second Edition. Oxford University Press, 2017.
Mitsuru Matsui . Linear cryptanalysis method for DES cipher. Lecture Notes in Computer Science, 765 (1994), 386–397. (EUROCRYPT ’93.)
Tsutomu Matsumoto , Youichi Takashima , and Hideki Imai . On seeking smart public-key distribution systems. Transactions of the IECE (Japan), 69 (1986), 99–106.
Tsutomu Matsumoto and Hideki Imai . Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. Lecture Notes in Computer Science, 330 (1988), 419–453. (EUROCRYPT ’88.)
Robert McEliece . A public-key cryptosystem based on algebraic coding theory. DSN Progress Report 42-44 (1978), 114–116.
Willi Meier and Othmar Staffelbach . Fast correlation attacks on certain stream ciphers. Journal of Cryptology 1 (1989) 159–176.
Alfred. J. Menezes , Tatsuaki Okamoto , and Scott A. Vanstone . Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory, 39 (1993), 1639–1646.
Alfred J. Menezes , Paul C. van Oorschot , and Scott A. Vanstone . Handbook of Applied Cryptography. CRC Press, 1996.
Ralph Merkle . Secure communications over insecure channels. Communications of the ACM, 21 (1978), 294–299.
Ralph Merkle . A certified digital signature. Lecture Notes in Computer Science, 435 (1990), 218–238. (CRYPTO ’89.)
Ralph Merkle . One way hash functions and DES. Lecture Notes in Computer Science, 435 (1990), 428–446. (CRYPTO ’89.)
Gary Miller . Riemann’s hypothesis and tests for primality. Journal of Computer and Systems Science, 13 (1976), 300–317.
Victor Miller . Uses of elliptic curves in cryptography. Lecture Notes in Computer Science, 218 (1986), 417–426. (CRYPTO ’85.)
Chris Mitchell , Fred Piper , and Peter Wild . Digital signatures. In Contemporary Cryptology, The Science of Information Integrity, pages 325–378. IEEE Press, 1992.
Judy Moore . Protocol failures in cryptosystems. In Contemporary Cryptology, The Science of Information Integrity, pages 541–558. IEEE Press, 1992.
Michele Mosca . Cybersecurity in an era with quantum computers: will we be ready? IACR ePrint Archive, report # 2015/1075. https://eprint.iacr.org/2015/1075.pdf
Gary Mullen and Daniel Pan ario , Eds. Handbook of Finite Fields. Chapman and Hall/CRC, 2013.
Satoshi Nakamoto . Bitcoin: a peer-to-peer electronic cash system. White paper, October 31, 2008. https://bitcoin.org/bitcoin.pdf
Moni Naor and Adi Shamir . Visual cryptography. Lecture Notes in Computer Science, 950 (1995), 1–12. (EUROCRYPT ’94.)
National Institute of Standards and Technology . Data Encryption Standard (DES). Federal Information Processing Standard (FIPS) Publication 46-3, October 1999. (Withdrawn on May 19, 2005.)
National Institute of Standards and Technology . Digital Signature Standard. Federal Information Processing Standard (FIPS) Publication 186-4, July 2013.
National Institute of Standards and Technology . Entity Authentication Using Public Key Cryptography. Federal Information Processing Standard (FIPS) Publication 196, February 1997. (Withdrawn on October 19, 2015.)
National Institute of Standards and Technology. Advanced Encryption Standard. Federal Information Processing Standard (FIPS) Publication 197, 2001.
National Institute of Standards and Technology. The Keyed-Hash Message Authentication Code (HMAC). Federal Information Processing Standard (FIPS) Publication 198-1, 2008.
National Institute of Standards and Technology. Secure Hash Standard (SHS). Federal Information Processing Standard (FIPS) Publication 180-4, 2015.
National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Federal Information Processing Standard (FIPS) Publication 202, 2015.
Vasilii Nechaev . On the complexity of a deterministic algorithm for a discrete logarithm. Math. Zametki, 55 (1994), 91–101.
Roger Needham and Michael Schroeder . Using encryption for authentication in large networks of computers. Communications of the ACM, 21 (1978), 993–999.
Michael Nielsen . How the Bitcoin protocol actually works. December 6,2013. http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/
Phong Nguyen and Igor Shparlinski . The insecurity of the digital signature algorithm with partially known nonces. Journal of Cryptology, 15 (2002), 151–176.
Christof Paar and Jan Pelzl . Understanding Cryptography: A Textbook for Students and Practitioners. Springer, 2010.
Jacques Patarin . Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. Lecture Notes in Computer Science, 1070 (1996), 33–48. (EUROCRYPT ’96.)
Jacques Patarin . The Oil and Vinegar signature scheme. Presented at the Dagstuhl Workshop on Cryptography, September 1997.
Rene Peralta . Simultaneous security of bits in the discrete log. Lecture Notes in Computer Science, 219 (1986), 62–72. (EUROCRYPT ’85.)
Pascal Paillier . Public-Key cryptosystems based on composite degree residuosity classes. Lecture Notes in Computer Science, 1592 (1999), 223–238. (EUROCRYPT ’99.)
Trevor Perrin (editor) and Moxie Marlinspike . The double ratchet algorithm. Open Whisper Systems, November 20, 2016. https://signal.org/docs/specifications/doubleratchet/
Stephen Pohlig and Martin Hellman . An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. IEEE Transactions on Information Theory, 24 (1978), 106–110.
David Pointcheval and Jacques Stern . Security arguments for signature schemes and blind signatures. Journal of Cryptology, 13 (2000), 361–396.
John Pollard . Monte Carlo methods for index computation (mod p). Mathematics of Computation, 32 (1978), 918–924.
Bart Preneel . The state of cryptographic hash functions. Lecture Notes in Computer Science, 1561 (1999), 158–182. (Lectures on Data Security.)

Michael Rabin . Digitized signatures and public-key functions as intractable as factorization. MIT Laboratory for Computer Science Technical Report, LCS/TR-212, 1979.
Michael Rabin . Probabilistic algorithms for testing primality. Journal of Number Theory, 12 (1980), 128–138.
Oded Regev . The learning with errors problem, In 25th IEEE Conference on Computational Complexity, pages 191–204. IEEE, 2010.
Ronald Rivest . The MD4 message digest algorithm. Lecture Notes in Computer Science, 537 (1991), 303–311. (CRYPTO ’90.)
Ronald Rivest . The MD5 message digest algorithm. Internet Network Working Group RFC 1321, April 1992.
Ronald Rivest , Adi Shamir , and Leonard Adleman . A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21 (1978), 120–126.
Arto Salomaa . Public-Key Cryptography. Springer, 1990.
Claus Schnorr . Efficient signature generation by smart cards. Journal of Cryptology, 4 (1991), 161–174.
Adi Shamir . How to share a secret. Communications of the ACM, 22 (1979), 612–613.
Adi Shamir . Identity-based cryptosystems and signature schemes. Lecture Notes in Computer Science, 196 (1985), 47–53. (CRYPTO ’84.)
Claude E. Shannon . A mathematical theory of communication. Bell Systems Technical Journal, 27 (1948), 379–423, 623–656.
Claude E. Shannon . Communication theory of secrecy systems. Bell Systems Technical Journal, 28 (1949), 656–715.
Victor Shoup . Lower bounds for discrete logarithms and related problems. Lecture Notes in Computer Science, 1233 (1997), 256–266. (EUROCRYPT ’97.)
Dan Shumow and Neils Ferguson . On the possibility of a back door in the NIST SP800-90 Dual Ec Prng. CRYPTO 2007 Rump Session. http://rump2007.cr.yp.to/15-shumow.pdf
Thomas Siegenthaler . Decrypting a class of stream ciphers using ciphertext only, IEEE Transactions on Computers 34 (1985), 81–85.
Gustavus J. Simmons . A survey of information authentication. In Contemporary Cryptology, The Science of Information Integrity, pages 379–419. IEEE Press, 1992.
Simon Singh . The Code Book: The Science Of Secrecy From Ancient Egypt To Quantum Cryptography. Anchor Books, 2000.
Nigel Smart . The discrete logarithm problem on elliptic curves of trace one. Journal of Cryptology, 12 (1999), 193–196.
Nigel Smart . Cryptography Made Simple. Springer, 2015.
Clayton D. Smith . Digital Signcryption. Masters Thesis, Department of Combinatorics and Optimization, University of Waterloo, 2005.
Jerome Solinas . Efficient arithmetic on Koblitz curves. Designs, Codes and Cryptography, 19 (2000), 195–249.
Robert Solovay and Volker Strassen . A fast Monte Carlo test for primality. SIAM Journal on Computing, 6 (1977), 84–85.
Jessica Staddon , Douglas R. Stinson , and Ruizhong Wei . Combinatorial properties of frameproof and traceability codes. IEEE Transactions on Information Theory, 47 (2001), 1042–1049.
Michael Steiner , Gene Tsudik , and Michael Waidner . Diffie-Hellman key distribution extended to group communication. In Proceedings of the 3rd ACM Conference on Computer and Communications Security, pages 31–37. ACM Press, 1996.
Marc Stevens , Elie Bursztein , Pierre Karpman , Ange Albertini , and Yarik Markov . The first collision for full SHA-1. Lecture Notes in Computer Science, 10401 (2017), 570–596. (Crypto 2017, Part I.)
Douglas Stinson . Some observations on the theory of cryptographic hash functions. Designs, Codes and Cryptography, 38 (2006), 259–277.
Chengdong Tao , Adama Diene , Shaohua Tang , and Jintai Ding . Simple matrix scheme for encryption. Lecture Notes in Computer Science, 7932 (2013), 231–242. (PQCrypto 2013.)
Edlyn Teske . On random walks for Pollard’s rho method. Mathematics of Computation, 70 (2001), 809–825.
Serge Vaudenay . Security flaws induced by CBC padding—Applications to SSL, IPSEC, WTLS .... Lecture Notes in Computer Science, 2332 (2002), 534–545. (EUROCRYPT 2002.)
Serge Vaudenay . A Classical Introduction to Cryptography: Applications for Communications Security. Springer, 2005.
Debby M. Wallner , Eric J. Harder , and Ryan C. Agee . Key management for multicast: issues and architectures. Internet Request for Comments 2627, June, 1999.
Lawrence Washington . Elliptic Curves: Number Theory and Cryptography, Second Edition. Chapman & Hall/CRC, 2008.
Mark Wegman and J. Lawrence Carter . New hash functions and their use in authentication and set equality. Journal of Computer and System Sciences, 22 (1981), 265–279.
Dominic Welsh . Codes and Cryptography. Oxford Science Publications, 1988.
Michael Wiener . Cryptanalysis of short RSA secret exponents. IEEE Transactions on Information Theory, 36 (1990), 553–558.
Hugh Williams . A modification of the RSA public-key encryption procedure. IEEE Transactions on Information Theory, 26 (1980), 726–729.
Chung Kei Wong and Simon S. Lam . Digital signatures for flows and multicasts. IEEE/ACM Transactions on Networking, 7 (1999), 502–513.
Tao Xie , Fanbao Liu , and Dengguo Feng . Fast Collision Attack on MD5. IACR ePrint Archive, report # 2013/170.
Song Yan . Cryptanalytic Attacks on RSA. Springer, 2008.
Andrew Yao . Theory and applications of trapdoor functions. In Proceedings of the 23rd Annual Symposium on the Foundations of Computer Science, pages 80–91. IEEE Press, 1982.
Yuliang Zheng . Digital signcryption or how to achieve cost(signature & encryption) cost(signature) + cost(encryption). Lecture Notes in Computer Science, 1294 (1997), 165–179. (CRYPTO ’97.)
Discrete logarithm records . https://en.wikipedia.org/wiki/Discrete_logarithm_records