HANDBOOK of APPLIED CRYPTOGRAPHY

Alfred J. Menezes
Paul C. van Oorschot
Scott A. Vanstone

Foreword

by R.L. Rivest

As we draw near to closing out the twentieth century, we see quite clearly that the information-processing and telecommunications revolutions now underway will continue vigorously into the twenty-first. We interact and transact by directing flocks of digital packets towards each other through cyberspace, carrying love notes, digital cash, and secret corporate documents. Our personal and economic lives rely more and more on our ability to let such ethereal carrier pigeons mediate at a distance what we used to do with face-to-face meetings, paper documents, and a firm handshake. Unfortunately, the technical wizardry enabling remote collaborations is founded on broadcasting everything as sequences of zeros and ones that one's own dog wouldn't recognize. What is to distinguish a digital dollar when it is as easily reproducible as the spoken word? How do we converse privately when every syllable is bounced off a satellite and smeared over an entire continent? How should a bank know that it really *is* Bill Gates requesting from his laptop in Fiji a transfer of $10,000,000,000 to another bank? Fortunately, the magical mathematics of cryptography can help. Cryptography provides techniques for keeping information secret, for determining that information has not been tampered with, and for determining who authored pieces of information.

Cryptography is fascinating because of the close ties it forges between theory and practice, and because today's practical applications of cryptography are pervasive and critical components of our information-based society. Information-protection protocols designed on theoretical foundations one year appear in products and standards documents the next. Conversely, new theoretical developments sometimes mean that last year's proposal has a previously unsuspected weakness. While the theory is advancing vigorously, there are as yet few true guarantees; the security of many proposals depends on unproven (if plausible) assumptions. The theoretical work refines and improves the practice, while the practice challenges and inspires the theoretical work. When a system is "broken," our knowledge improves, and next year's system is improved to repair the defect. (One is reminded of the long and intriguing battle between the designers of bank vaults and their opponents.)

Cryptography is also fascinating because of its game-like adversarial nature. A good cryptographer rapidly changes sides back and forth in his or her thinking, from attacker to defender and back. Just as in a game of chess, sequences of moves and counter-moves must be considered until the current situation is understood. Unlike chess players, cryptographers must also consider all the ways an adversary might try to gain by breaking the rules or violating expectations. (Does it matter if she measures how long I am computing? Does it matter if her "random" number isn't one?)

The current volume is a major contribution to the field of cryptography. It is a rigorous encyclopedia of known techniques, with an emphasis on those that are both (believed to be) secure and practically useful. It presents in a coherent manner most of the important cryptographic tools one needs to implement secure cryptographic systems, and explains many of the cryptographic principles and protocols of existing systems. The topics covered range from low-level considerations such as random-number generation and efficient modular exponentiation algorithms and medium-level items such as public-key signature techniques, to higher-level topics such as zero-knowledge protocols. This book's excellent organization and style allow it to serve well as both a self-contained tutorial and an indispensable desk reference.

In documenting the state of a fast-moving field, the authors have done incredibly well at providing error-free comprehensive content that is up-to-date. Indeed, many of the chapters, such as those on hash functions or key-establishment protocols, break new ground in both their content and their unified presentations. In the trade-off between comprehensive coverage and exhaustive treatment of individual items, the authors have chosen to write simply and directly, and thus efficiently, allowing each element to be explained together with their important details, caveats, and comparisons.

While motivated by practical applications, the authors have clearly written a book that will be of as much interest to researchers and students as it is to practitioners, by including ample discussion of the underlying mathematics and associated theoretical considerations. The essential mathematical techniques and requisite notions are presented crisply and clearly, with illustrative examples. The insightful historical notes and extensive bibliography make this book a superb stepping-stone to the literature. (I was very pleasantly surprised to find an appendix with complete programs for the CRYPTO and EUROCRYPT conferences!)

It is a pleasure to have been asked to provide the foreword for this book. I am happy to congratulate the authors on their accomplishment, and to inform the reader that he/she is looking at a landmark in the development of the field.

Ronald L. Rivest
Webster Professor of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
June 1996

Preface

This book is intended as a reference for professional cryptographers, presenting the techniques and algorithms of greatest interest to the current practitioner, along with the supporting motivation and background material. It also provides a comprehensive source from which to learn cryptography, serving both students and instructors. In addition, the rigorous treatment, breadth, and extensive bibliographic material should make it an important reference for research professionals.

Our goal was to assimilate the existing cryptographic knowledge of industrial interest into one consistent, self-contained volume accessible to engineers in practice, to computer scientists and mathematicians in academia, and to motivated non-specialists with a strong desire to learn cryptography. Such a task is beyond the scope of each of the following: research papers, which by nature focus on narrow topics using very specialized (and often non-standard) terminology; survey papers, which typically address, at most, a small number of major topics at a high level; and (regretably also) most books, due to the fact that many book authors lack either practical experience or familiarity with the research literature or both. Our intent was to provide a detailed presentation of those areas of cryptography which we have found to be of greatest practical utility in our own industrial experience, while maintaining a sufficiently formal approach to be suitable both as a trustworthy reference for those whose primary interest is further research, and to provide a solid foundation for students and others first learning the subject.

Throughout each chapter, we emphasize the relationship between various aspects of cryptography. Background sections commence most chapters, providing a framework and perspective for the techniques which follow. Computer source code (e.g. C code) for algorithms has been intentionally omitted, in favor of algorithms specified in sufficient detail to allow direct implementation without consulting secondary references. We believe this style of presentation allows a better understanding of how algorithms actually work, while at the same time avoiding low-level implementation-specific constructs (which some readers will invariably be unfamiliar with) of various currently-popular programming languages.

The presentation also strongly delineates what has been established as fact (by mathematical arguments) from what is simply current conjecture. To avoid obscuring the very applied nature of the subject, rigorous proofs of correctness are in most cases omitted; however, references given in the Notes section at the end of each chapter indicate the original or recommended sources for these results. The trailing Notes sections also provide information (quite detailed in places) on various additional techniques not addressed in the main text, and provide a survey of research activities and theoretical results; references again indicate where readers may pursue particular aspects in greater depth. Needless to say, many results, and indeed some entire research areas, have been given far less attention than they warrant, or have been omitted entirely due to lack of space; we apologize in advance for such major omissions, and hope that the most significant of these are brought to our attention.

To provide an integrated treatment of cryptography spanning foundational motivation through concrete implementation, it is useful to consider a hierarchy of thought ranging from conceptual ideas and end-user services, down to the tools necessary to complete actual implementations. Table 1 depicts the hierarchical structure around which this book is organized. Corresponding to this, Figure 1 illustrates how these hierarchical levels map

Information Security Objectives	
Confidentiality	
Data integrity	
Authentication (entity and data origin)	
Non-repudiation	
Cryptographic functions	
Encryption	Chapters 6, 7, 8
Message authentication and data integrity techniques	Chapter 9
Identification/entity authentication techniques	Chapter 10
Digital signatures	Chapter 11
Cryptographic building blocks	
Stream ciphers	Chapter 6
Block ciphers (symmetric-key)	Chapter 7
Public-key encryption	Chapter 8
One-way hash functions (unkeyed)	Chapter 9
Message authentication codes	Chapter 9
Signature schemes (public-key, symmetric-key)	Chapter 11
Utilities	
Public-key parameter generation	Chapter 4
Pseudorandom bit generation	Chapter 5
Efficient algorithms for discrete arithmetic	Chapter 14
Foundations	
Introduction to cryptography	Chapter 1
Mathematical background	Chapter 2
Complexity and analysis of underlying problems	Chapter 3
Infrastructure techniques and commercial aspects	
Key establishment protocols	Chapter 12
Key installation and key management	Chapter 13
Cryptographic patents	Chapter 15
Cryptographic standards	Chapter 15

Table 1: *Hierarchical levels of applied cryptography.*

onto the various chapters, and their inter-dependence.

Table 2 lists the chapters of the book, along with the primary author(s) of each who should be contacted by readers with comments on specific chapters. Each chapter was written to provide a self-contained treatment of one major topic. Collectively, however, the chapters have been designed and carefully integrated to be entirely complementary with respect to definitions, terminology, and notation. Furthermore, there is essentially no duplication of material across chapters; instead, appropriate cross-chapter references are provided where relevant.

While it is not intended that this book be read linearly from front to back, the material has been arranged so that doing so has some merit. Two primary goals motivated by the "handbook" nature of this project were to allow easy access to stand-alone results, and to allow results and algorithms to be easily referenced (e.g., for discussion or subsequent cross-reference). To facilitate the ease of accessing and referencing results, items have been categorized and numbered to a large extent, with the following classes of items jointly numbered consecutively in each chapter: *Definitions*, *Examples*, *Facts*, *Notes*, *Remarks*, *Algorithms*, *Protocols*, and *Mechanisms*. In more traditional treatments, *Facts* are usually identified as propositions, lemmas, or theorems. We use numbered *Notes* for additional technical points,

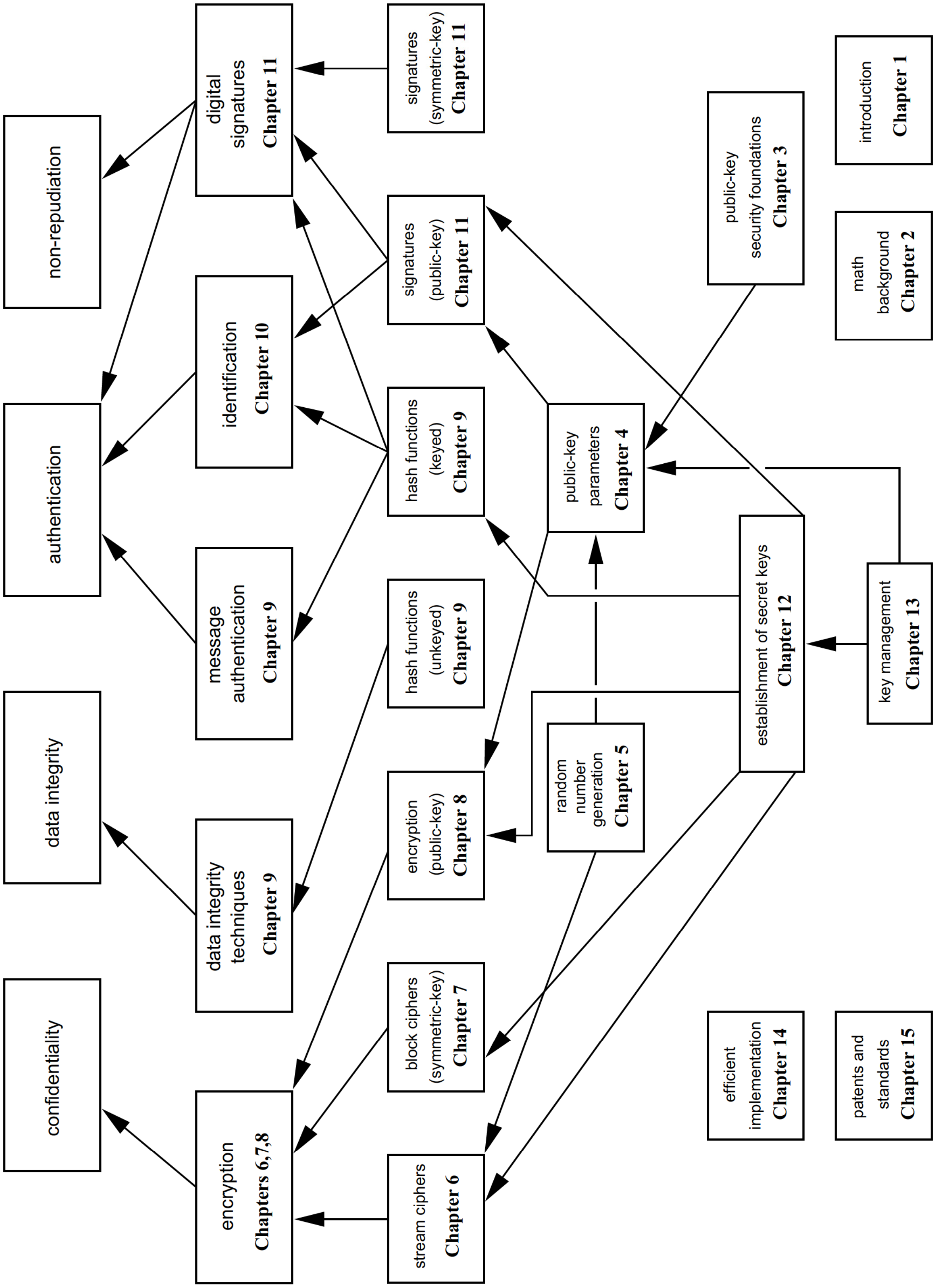

Figure 1: *Roadmap of the book.*

	Chapter	Primary Author		
		AJM	PVO	SAV
1.	Overview of Cryptography	*	*	*
2.	Mathematical Background	*		
3.	Number-Theoretic Reference Problems	*		
4.	Public-Key Parameters	*	*	
5.	Pseudorandom Bits and Sequences	*		
6.	Stream Ciphers	*		
7.	Block Ciphers		*	
8.	Public-Key Encryption	*		
9.	Hash Functions and Data Integrity		*	
10.	Identification and Entity Authentication		*	
11.	Digital Signatures			*
12.	Key Establishment Protocols		*	
13.	Key Management Techniques		*	
14.	Efficient Implementation			*
15.	Patents and Standards		*	
—	Overall organization	*	*	

Table 2: *Primary authors of each chapter.*

while numbered *Remarks* identify non-technical (often non-rigorous) comments, observations, and opinions. *Algorithms*, *Protocols* and *Mechanisms* refer to techniques involving a series of steps. *Examples*, *Notes*, and *Remarks* generally begin with parenthetical summary titles to allow faster access, by indicating the nature of the content so that the entire item itself need not be read in order to determine this. The use of a large number of small subsections is also intended to enhance the handbook nature and accessibility to results.

Regarding the partitioning of subject areas into chapters, we have used what we call a *functional organization* (based on functions of interest to end-users). For example, all items related to entity authentication are addressed in one chapter. An alternative would have been what may be called an *academic organization*, under which perhaps, all protocols based on zero-knowledge concepts (including both a subset of entity authentication protocols and signature schemes) might be covered in one chapter. We believe that a functional organization is more convenient to the practitioner, who is more likely to be interested in options available for an entity authentication protocol (Chapter 10) or a signature scheme (Chapter 11), than to be seeking a zero-knowledge protocol with unspecified end-purpose.

In the front matter, a top-level Table of Contents (giving chapter numbers and titles only) is provided, as well as a detailed Table of Contents (down to the level of subsections, e.g., §5.1.1). This is followed by a List of Figures, and a List of Tables. At the start of each chapter, a brief Table of Contents (specifying section number and titles only, e.g., §5.1, §5.2) is also given for convenience.

At the end of the book, we have included a list of papers presented at each of the Crypto, Eurocrypt, Asiacrypt/Auscrypt and Fast Software Encryption conferences to date, as well as a list of all papers published in the *Journal of Cryptology* up to Volume 9. These are in addition to the *References* section, each entry of which is cited at least once in the body of the handbook. Almost all of these references have been verified for correctness in their exact titles, volume and page numbers, etc. Finally, an extensive Index prepared by the authors is included. The Index begins with a List of Symbols.

Our intention was not to introduce a collection of new techniques and protocols, but rather to selectively present techniques from those currently available in the public domain. Such a consolidation of the literature is necessary from time to time. The fact that many good books in this field include essentially no more than what is covered here in Chapters 7, 8 and 11 (indeed, these might serve as an introductory course along with Chapter 1) illustrates that the field has grown tremendously in the past 15 years. The mathematical foundation presented in Chapters 2 and 3 is hard to find in one volume, and missing from most cryptography texts. The material in Chapter 4 on generation of public-key parameters, and in Chapter 14 on efficient implementations, while well-known to a small body of specialists and available in the scattered literature, has previously not been available in general texts. The material in Chapters 5 and 6 on pseudorandom number generation and stream ciphers is also often absent (many texts focus entirely on block ciphers), or approached only from a theoretical viewpoint. Hash functions (Chapter 9) and identification protocols (Chapter 10) have only recently been studied in depth as specialized topics on their own, and along with Chapter 12 on key establishment protocols, it is hard to find consolidated treatments of these now-mainstream topics. Key management techniques as presented in Chapter 13 have traditionally not been given much attention by cryptographers, but are of great importance in practice. A focused treatment of cryptographic patents and a concise summary of cryptographic standards, as presented in Chapter 15, are also long overdue.

In most cases (with some historical exceptions), where algorithms are known to be insecure, we have chosen to leave out specification of their details, because most such techniques are of little practical interest. Essentially all of the algorithms included have been verified for correctness by independent implementation, confirming the test vectors specified.

Acknowledgements

This project would not have been possible without the tremendous efforts put forth by our peers who have taken the time to read endless drafts and provide us with technical corrections, constructive feedback, and countless suggestions. In particular, the advice of our Advisory Editors has been invaluable, and it is impossible to attribute individual credit for their many suggestions throughout this book. Among our Advisory Editors, we would particularly like to thank:

Mihir Bellare
Don Coppersmith
Dorothy Denning
Walter Fumy
Burt Kaliski
Peter Landrock
Arjen Lenstra
Ueli Maurer
Chris Mitchell
Tatsuaki Okamoto
Bart Preneel
Ron Rivest
Gus Simmons
Miles Smid
Jacques Stern
Mike Wiener
Yacov Yacobi

In addition, we gratefully acknowledge the exceptionally large number of additional individuals who have helped improve the quality of this volume, by providing highly appreciated feedback and guidance on various matters. These individuals include:

Carlisle Adams
Rich Ankney
Tom Berson
Simon Blackburn
Ian Blake
Antoon Bosselaers
Colin Boyd
Jörgen Brandt
Mike Burmester
Ed Dawson
Peter de Rooij
Yvo Desmedt
Whit Diffie
Hans Dobbertin
Carl Ellison
Luis Encinas
Warwick Ford
Amparo Fuster
Shuhong Gao
Will Gilbert
Marc Girault
Jovan Golić
Dieter Gollmann
Li Gong

Carrie Grant
Blake Greenlee
Helen Gustafson
Darrel Hankerson
Anwar Hasan
Don Johnson
Mike Just
Andy Klapper
Lars Knudsen
Neal Koblitz
Çetin Koç
Judy Koeller
Evangelos Kranakis
David Kravitz
Hugo Krawczyk
Xuejia Lai
Charles Lam
Alan Ling
S. Mike Matyas
Willi Meier
Peter Montgomery
Mike Mosca
Tim Moses
Serge Mister
Volker Müeller
David Naccache
James Nechvatal
Kaisa Nyberg
Andrew Odlyzko
Richard Outerbridge
Walter Penzhorn
Birgit Pfitzmann
Kevin Phelps
Leon Pintsov
Fred Piper
Carl Pomerance
Matt Robshaw
Peter Rodney
Phil Rogaway
Rainer Rueppel
Mahmoud Salmasizadeh
Roger Schlafly
Jeff Shallit
Jon Sorenson
Doug Stinson
Andrea Vanstone
Serge Vaudenay
Klaus Vedder
Jerry Veeh
Fausto Vitini
Lisa Yin
Robert Zuccherato

We apologize to those whose names have inadvertently escaped this list. Special thanks are due to Carrie Grant, Darrel Hankerson, Judy Koeller, Charles Lam, and Andrea Vanstone. Their hard work contributed greatly to the quality of this book, and it was truly a pleasure working with them. Thanks also to the folks at CRC Press, including Tia Atchison, Gary Bennett, Susie Carlisle, Nora Konopka, Mary Kugler, Amy Morrell, Tim Pletscher, Bob Stern, and Wayne Yuhasz. The second author would like to thank his colleagues past and present at Nortel Secure Networks (Bell-Northern Research), many of whom are mentioned above, for their contributions on this project, and in particular Brian O'Higgins for his encouragement and support; all views expressed, however, are entirely that of the author. The third author would also like to acknowledge the support of the Natural Sciences and Engineering Research Council.

Any errors that remain are, of course, entirely our own. We would be grateful if readers who spot errors, missing references or credits, or incorrectly attributed results would contact us with details. It is our hope that this volume facilitates further advancement of the field, and that we have helped play a small part in this.

Alfred J. Menezes
Paul C. van Oorschot
Scott A. Vanstone
August, 1996

Table of Contents

Chapter 6

Stream Ciphers

Contents in Brief

6.1 Introduction

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. By contrast, *block ciphers* (Chapter 7) tend to simultaneously encrypt groups of characters of a plaintext message using a fixed encryption transformation. Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry. They are also more appropriate, and in some cases mandatory (e.g., in some telecommunications applications), when buffering is limited or when characters must be individually processed as they are received. Because they have limited or no error propagation, stream ciphers may also be advantageous in situations where transmission errors are highly probable.

There is a vast body of theoretical knowledge on stream ciphers, and various design principles for stream ciphers have been proposed and extensively analyzed. However, there are relatively few fully-specified stream cipher algorithms in the open literature. This unfortunate state of affairs can partially be explained by the fact that most stream ciphers used in practice tend to be proprietary and confidential. By contrast, numerous concrete block cipher proposals have been published, some of which have been standardized or placed in the public domain. Nevertheless, because of their significant advantages, stream ciphers are widely used today, and one can expect increasingly more concrete proposals in the coming years.

Chapter outline

The remainder of §6.1 introduces basic concepts relevant to stream ciphers. Feedback shift registers, in particular linear feedback shift registers (LFSRs), are the basic building block in most stream ciphers that have been proposed; they are studied in §6.2. Three general techniques for utilizing LFSRs in the construction of stream ciphers are presented in §6.3: using a nonlinear combining function on the outputs of several LFSRs (§6.3.1), using a nonlinear filtering function on the contents of a single LFSR (§6.3.2), and using the output of one (or more) LFSRs to control the clock of one (or more) other LFSRs (§6.3.3). Two concrete proposals for clock-controlled generators, the alternating step generator and the shrinking generator are presented in §6.3.3. §6.4 presents a stream cipher not based on LFSRs, namely SEAL. §6.5 concludes with references and further chapter notes.

6.1.1 Classification

Stream ciphers can be either symmetric-key or public-key. The focus of this chapter is symmetric-key stream ciphers; the Blum-Goldwasser probabilistic public-key encryption scheme (§8.7.2) is an example of a public-key stream cipher.

6.1 Note (*block vs. stream ciphers*) Block ciphers process plaintext in relatively large blocks (e.g., $n \geq 64$ bits). The same function is used to encrypt successive blocks; thus (pure) block ciphers are *memoryless*. In contrast, stream ciphers process plaintext in blocks as small as a single bit, and the encryption function may vary as plaintext is processed; thus stream ciphers are said to have memory. They are sometimes called *state ciphers* since encryption depends on not only the key and plaintext, but also on the current state. This distinction between block and stream ciphers is not definitive (see Remark 7.25); adding a small amount of memory to a block cipher (as in the CBC mode) results in a stream cipher with large blocks.

(i) The one-time pad

Recall (Definition 1.39) that a *Vernam cipher* over the binary alphabet is defined by

$$c_i = m_i \oplus k_i \quad \text{for } i = 1, 2, 3 \ldots ,$$

where $m_1, m_2, m_3, \ldots$ are the plaintext digits, $k_1, k_2, k_3, \ldots$ (the *keystream*) are the key digits, $c_1, c_2, c_3, \ldots$ are the ciphertext digits, and $\oplus$ is the XOR function (bitwise addition modulo 2). Decryption is defined by $m_i = c_i \oplus k_i$. If the keystream digits are generated independently and randomly, the Vernam cipher is called a *one-time pad*, and is unconditionally secure (§1.13.3(i)) against a ciphertext-only attack. More precisely, if M, C, and K are random variables respectively denoting the plaintext, ciphertext, and secret key, and if $H()$ denotes the entropy function (Definition 2.39), then $H(M|C) = H(M)$. Equivalently, $I(M;C) = 0$ (see Definition 2.45): the ciphertext contributes no information about the plaintext.

Shannon proved that a necessary condition for a symmetric-key encryption scheme to be unconditionally secure is that $H(K) \geq H(M)$. That is, the uncertainty of the secret key must be at least as great as the uncertainty of the plaintext. If the key has bitlength k, and the key bits are chosen randomly and independently, then $H(K) = k$, and Shannon's necessary condition for unconditional security becomes $k \geq H(M)$. The one-time pad is unconditionally secure regardless of the statistical distribution of the plaintext, and is optimal in the sense that its key is the smallest possible among all symmetric-key encryption schemes having this property.

An obvious drawback of the one-time pad is that the key should be as long as the plaintext, which increases the difficulty of key distribution and key management. This motivates the design of stream ciphers where the keystream is *pseudorandomly* generated from a smaller secret key, with the intent that the keystream appears random to a computationally bounded adversary. Such stream ciphers do not offer unconditional security (since $H(K) \ll H(M)$), but the hope is that they are computationally secure (§1.13.3(iv)).

Stream ciphers are commonly classified as being *synchronous* or *self-synchronizing*.

(ii) Synchronous stream ciphers

6.2 Definition A *synchronous* stream cipher is one in which the keystream is generated independently of the plaintext message and of the ciphertext.

The encryption process of a synchronous stream cipher can be described by the equations

$$\begin{aligned} \sigma_{i+1} &= f(\sigma_i, k), \\ z_i &= g(\sigma_i, k), \\ c_i &= h(z_i, m_i), \end{aligned}$$

where σ_0 is the *initial state* and may be determined from the key k, f is the *next-state function*, g is the function which produces the *keystream* z_i, and h is the *output function* which combines the keystream and plaintext m_i to produce ciphertext c_i. The encryption and decryption processes are depicted in Figure 6.1. The OFB mode of a block cipher (see §7.2.2(iv)) is an example of a synchronous stream cipher.

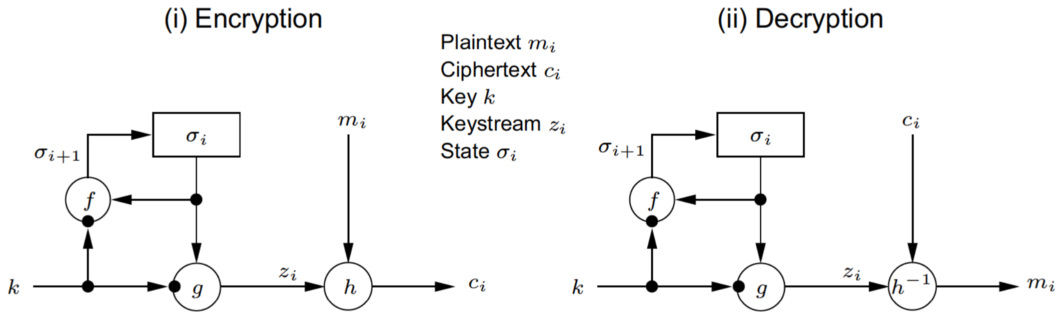

Figure 6.1: *General model of a synchronous stream cipher.*

6.3 Note (*properties of synchronous stream ciphers*)

(i) *synchronization requirements.* In a synchronous stream cipher, both the sender and receiver must be *synchronized* – using the same key and operating at the same position (state) within that key – to allow for proper decryption. If synchronization is lost due to ciphertext digits being inserted or deleted during transmission, then decryption fails and can only be restored through additional techniques for re-synchronization. Techniques for re-synchronization include re-initialization, placing special markers at regular intervals in the ciphertext, or, if the plaintext contains enough redundancy, trying all possible keystream offsets.

(ii) *no error propagation.* A ciphertext digit that is modified (but not deleted) during transmission does not affect the decryption of other ciphertext digits.

(iii) *active attacks.* As a consequence of property (i), the insertion, deletion, or replay of ciphertext digits by an active adversary causes immediate loss of synchronization, and hence might possibly be detected by the decryptor. As a consequence of property (ii), an active adversary might possibly be able to make changes to selected ciphertext digits, and know exactly what affect these changes have on the plaintext. This illustrates that additional mechanisms must be employed in order to provide data origin authentication and data integrity guarantees (see §9.5.4).

Most of the stream ciphers that have been proposed to date in the literature are additive stream ciphers, which are defined below.

6.4 Definition A *binary additive stream cipher* is a synchronous stream cipher in which the keystream, plaintext, and ciphertext digits are binary digits, and the output function h is the XOR function.

Binary additive stream ciphers are depicted in Figure 6.2. Referring to Figure 6.2, the *keystream generator* is composed of the next-state function f and the function g (see Figure 6.1), and is also known as the *running key generator*.

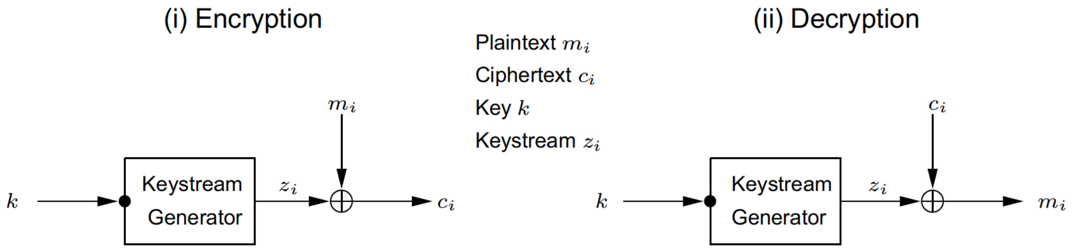

Figure 6.2: *General model of a binary additive stream cipher.*

(iii) Self-synchronizing stream ciphers

6.5 Definition A *self-synchronizing* or *asynchronous* stream cipher is one in which the keystream is generated as a function of the key and a fixed number of previous ciphertext digits.

The encryption function of a self-synchronizing stream cipher can be described by the equations

$$\begin{aligned}\sigma_i &= (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}),\\ z_i &= g(\sigma_i, k),\\ c_i &= h(z_i, m_i),\end{aligned}$$

where $\sigma_0 = (c_{-t}, c_{-t+1}, \ldots, c_{-1})$ is the (non-secret) *initial state*, k is the *key*, g is the function which produces the *keystream* z_i, and h is the *output function* which combines the keystream and plaintext m_i to produce ciphertext c_i. The encryption and decryption processes are depicted in Figure 6.3. The most common presently-used self-synchronizing stream ciphers are based on block ciphers in 1-bit cipher feedback mode (see §7.2.2(iii)).

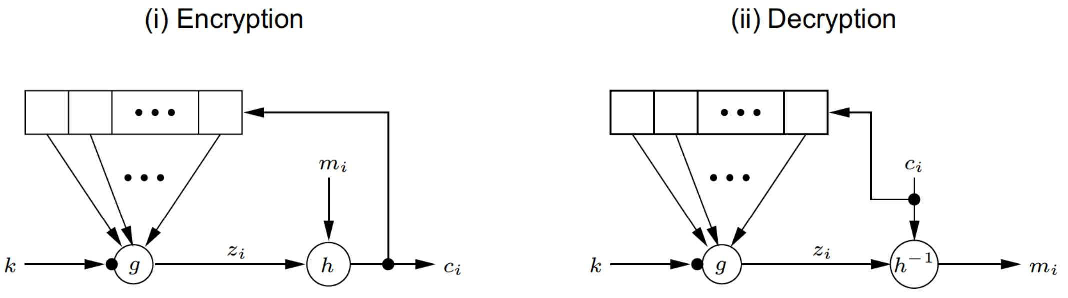

Figure 6.3: *General model of a self-synchronizing stream cipher.*

6.6 Note (*properties of self-synchronizing stream ciphers*)

(i) *self-synchronization.* Self-synchronization is possible if ciphertext digits are deleted or inserted, because the decryption mapping depends only on a fixed number of preceding ciphertext characters. Such ciphers are capable of re-establishing proper decryption automatically after loss of synchronization, with only a fixed number of plaintext characters unrecoverable.

(ii) *limited error propagation.* Suppose that the state of a self-synchronization stream cipher depends on t previous ciphertext digits. If a single ciphertext digit is modified (or even deleted or inserted) during transmission, then decryption of up to t subsequent ciphertext digits may be incorrect, after which correct decryption resumes.

(iii) *active attacks.* Property (ii) implies that any modification of ciphertext digits by an active adversary causes several other ciphertext digits to be decrypted incorrectly, thereby improving (compared to synchronous stream ciphers) the likelihood of being detected by the decryptor. As a consequence of property (i), it is more difficult (than for synchronous stream ciphers) to detect insertion, deletion, or replay of ciphertext digits by an active adversary. This illustrates that additional mechanisms must be employed in order to provide data origin authentication and data integrity guarantees (see §9.5.4).

(iv) *diffusion of plaintext statistics.* Since each plaintext digit influences the entire following ciphertext, the statistical properties of the plaintext are dispersed through the ciphertext. Hence, self-synchronizing stream ciphers may be more resistant than synchronous stream ciphers against attacks based on plaintext redundancy.

6.2 Feedback shift registers

Feedback shift registers, in particular linear feedback shift registers, are the basic components of many keystream generators. §6.2.1 introduces linear feedback shift registers. The linear complexity of binary sequences is studied in §6.2.2, while the Berlekamp-Massey algorithm for computing it is presented in §6.2.3. Finally, nonlinear feedback shift registers are discussed in §6.2.4.

6.2.1 Linear feedback shift registers

Linear feedback shift registers (LFSRs) are used in many of the keystream generators that have been proposed in the literature. There are several reasons for this:

1. LFSRs are well-suited to hardware implementation;
2. they can produce sequences of large period (Fact 6.12);
3. they can produce sequences with good statistical properties (Fact 6.14); and
4. because of their structure, they can be readily analyzed using algebraic techniques.

6.7 Definition A *linear feedback shift register* (LFSR) of length L consists of L *stages* (or *delay elements*) numbered $0, 1, \ldots, L-1$, each capable of storing one bit and having one input and one output; and a clock which controls the movement of data. During each unit of time the following operations are performed:

(i) the content of stage 0 is output and forms part of the *output sequence*;