Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

325

Gilles Brassard

Modern Cryptology

A Tutorial

现代密码学 [英]

Springer-Verlag
World Publishing Corp

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

325

Gilles Brassard

Modern Cryptology

A Tutorial

Springer-Verlag
World Publishing Corp

Author

Gilles Brassard
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succursale "A"
Montréal Québec, Canada H3C 3J7

Reprinted by World Publishing Corporation, Beijing, 1990
for distribution and sale in The People's Republic of China only
ISBN 7-5062-0771-0

CR Subject Classification (1988): E.3

ISBN 0-387-96842-3 Springer-Verlag New York Berlin Heidelberg
ISBN 3-540-96842-3 Springer-Verlag Berlin Heidelberg New York

PREFACE

Dédié à Manuel Blum,
Ami inestimable

The present work grew out of lecture notes I prepared for a 3½ hour tutorial that I was invited to give at the *29th IEEE Computer Conference (CompCon)* in San Francisco on February 27, 1987. I have just completed a substantial update of the material, including the addition of several topics. My main purpose is to provide a self-contained overview of recent cryptologic achievements in a form that can be understood by readers having no previous acquaintance with cryptology. It can thus be used as preliminary reading material for an introductory class. Nevertheless, it covers enough state-of-the-art material to be also of interest to the specialist. An extensive bibliography is included.

I was originally invited to give this *CompCon* tutorial by Russel Brand. I was subsequently encouraged by David Kahn and Ronald L. Rivest to "do something" with the notes I had produced. The idea of transforming them into a monograph for Springer-Verlag's *Lecture Notes in Computer Science* series was suggested by Lynn Montz.

Many people have generously given countless hours helping me with my undertaking. Bonnie Berger, David Chaum, Yvo Desmedt, Bennett Fox, Silvio Micali and Jean-Jacques Quisquater have provided significant feedback through several revisions. It is also a great pleasure to acknowledge the help of my co-authors for the following parts of this work: Claude Crépeau and Claude Goutier for section 5.3, David Chaum and Claude Crépeau for sections 5.5 and 5.6, and Charles H. Bennett for the whole of chapter 6. In addition, section 5.7 was written entirely by David Chaum.

David Chaum and Jean-Jacques Quisquater have made it possible for me to work on this project while I visited them at the Centrum voor Wiskunde en Informatica (CWI), Amsterdam, and Philips Research Laboratory, Bruxelles, respectively. I am also particularly indebted to Jurjen N.E. Bos for his unstinting and never failing help at the CWI. The final manuscript was produced on the CWI Harris typesetter using their Boring computer running troff and the UNIX operating system. My research activities are supported by Canada's NSERC grant A4107.

Although they are too numerous to list explicitly, I would like to thank all those with whom I had insightful and stimulating discussions on cryptologic matters over the years. They make it an active and exciting field to work in. Last but not least, I offer all my gratitude to my wife Isabelle. Not only did she put up with my working on revising and typesetting the *CompCon* notes *immediately after* I had just finished going through the agony of producing the English version of my textbook on *Algorithmics* [52], but she actually typed in most of the original *CompCon* notes on and around New Year's Day, 1987.

It is possible that I will one day pluck up the courage to transform this monograph into a full-length textbook. In order to help me avoid carrying over errors from the former to the latter, I would be grateful to anyone kind enough to point out to me mistake(s) he/she may find in the present work, including trivial typographical errors. I would also appreciate information about the existence of final papers for the references listed here as appearing in various proceedings. Please direct all mail to me at the following address: *Département d'informatique et de recherche opérationnelle, Université de Montréal, C.P. 6128, Succursale "A", Montréal (Québec), CANADA H3C 3J7*. Thank you all in advance.

Gilles Brassard
Bruxelles, April 1988

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

"A man is crazy who writes a secret in any other way than one which will conceal it from the vulgar." [9]

— Roger Bacon, *circa* 1250

For thousands of years, *cryptography* has been the art of providing secure communication over insecure channels, and *cryptanalysis* has been the dual art of breaking into such communications. Historically, *cryptology* (the combined art of cryptography and cryptanalysis) has been almost exclusively in the hands of the military and diplomats. With the advent of the computer revolution, and more importantly of a society in which vast amounts of personal, financial, commercial and technological information are stored in computer data banks and transferred over computer networks, the necessity for civilian cryptography has become overwhelming. To put it in the words of Kahn, "Cryptography, in 1945 a nation's most closely held secret, has gone public" [155]. (See also [160, 161, 162, 185].)

Who is going to win the age-old battle between cryptography and cryptanalysis? Great (non-specialist) minds of past centuries disagree. In his *Dictionnaire philosophique* (1769), Voltaire wrote: "Ceux qui se vantent de lire les lettres chiffrées sont de plus grands charlatans que ceux qui se vanteraient d'entendre une langue qu'ils n'ont point apprise" [235] (loosely translated, this says: "Cryptanalysts are a bunch of charlatans, even more so than Champollion!"). The opposite opinion is voiced by Edgar Allan Poe in his famous tale *The Gold-Bug* (1843): "It may well be doubted whether human ingenuity can construct an enigma of this kind [a cryptogram] which human ingenuity may not, by proper application, resolve." [190].

It is now clear that Voltaire was wrong: most historical cryptosystems have been badly broken, sometimes with spectacular consequences [154]. On the other hand, there are cryptosystems that have been proved to be unbreakable, regardless of the cryptanalyst's "ingenuity" or computing power (such as the *one time pad,* discussed in section 3.2). Nevertheless, the question remains wide open for the more practical *public-key* cryptosystems (which is the topic of chapter 4). The current belief is that the increase in computing power witnessed in the later parts of this century places cryptographers in an unprecedentedly favourable position, to the detriment of cryptanalysts. This is ironic because the Colossus, which was the very first electronic computer in history, was built for the specific purpose of cryptanalysing German ciphers [138, 200]. (Brian Randell is reported to have once said: "By my reckoning, ENIAC was not the first computer, it was the eleventh." [250].) Thus, it may be said that cryptanalysis, being the "midwife of computer science" (Ronald L. Rivest [204]), has perhaps bred the instrument of its own doom!

Until recently, the presumed reliability of cryptosystems was "attested" by the amount of effort spent by qualified cryptanalysts in their unsuccessful attempts at breaking them. History has clearly indicated the pitfalls of this approach, as messages enciphered by cryptosystems that were believed to be invulnerable by their users were routinely decrypted. The breaking of Enigma by the Allies during (and even before) World War II is a prime example of this situation [122, 201]. The reader interested in the historical significance of cryptology is encouraged to read Kahn's wonderful account [154] and other popular books such as [246, 64, 122, 156, 92, 242].

In this century, mathematicians have worked at finding objective criteria for the security of cryptosystems, thereby transforming this ancient art into an exact science. Shannon developed information theory [214] as a result of his previous (originally classified) work on cryptography [215]. For various cryptosystems, he was able to estimate the amount of ciphertext required for a crytanalysis to achieve any desired level of reliability. For instance, Ib Melchior could have saved a trip to Elsinore, had he only believed in Shannon's theory, when he thought he had decrypted a secret message on Shakespeare's tombstone revealing the existence of a first Hamlet edition [154].

In the past decade, computer scientists have worked at basing the security of cryptography on the more recent theory of computational complexity instead of Shannon's information theory [100, 101, 179]. The basic difference is that Shannon s theory lives on the hope that the cryptanalyst will not have enough *information* to decipher a cryptogram, while computational complexity only expects the cryptanalyst not to have enough *time* to do so.

The purpose of the present work is to give an overview of recent cryptographic achievements and techniques, and of their present and potential applications. No particular background is expected from the reader. Although the coverage of several topics is necessarily brief, an extensive (but of course not exhaustive) list of references is provided. In addition to the historical books mentioned previously, more technical books are available, such as [116, 163, 182, 94, 88, 165, 226, 194], as well as several popular [153, 111, 120, 147, 221, 51] and technical [103, 217, 169, 7, 167, 62, 142, 204] survey articles.

CHAPTER 2

DEFINITIONS AND CLASSIFICATION

The purpose of a *cryptosystem* is to *encipher* an intelligible *cleartext* (also called *plaintext*), thus producing an unintelligible *ciphertext* (also called *cryptogram*). The intended receiver must be able to *decipher* the ciphertext, thus recovering the cleartext. However, *eavesdroppers* (also called *cryptanalysts*) must be unable to *decrypt* the ciphertext. Notice the important difference between deciphering and decryption.

There are several ways in which cryptosystems can be classified. We consider the following as most fundamental:

- Restricted use cryptosystems
- General use cryptosystems
  - secret-key
  - public-key.

A cryptosystem is *restricted* if its security is based on keeping secret the nature of the enciphering and deciphering algorithms. The simplest historic such system is the so-called *Ceasar cipher*, which simply consists of replacing each letter in the plaintext with the third following letter in the alphabet (with wraparound when necessary). For instance, the word "cleartext" becomes "fohduwhaw". Restricted systems are usually designed by amateurs and are almost always child's play for professionally experienced cryptanalysts. More importantly, such systems are of no use in the modern context of a large number of users. *Codes*, which are instances of restricted cryptosystems, are not discussed here.

A cryptosystem is *general* if its security lies not in the secrecy of the enciphering and deciphering algorithms, but rather on a relatively-short secret value known as the *key*. It should be easy for individual users to come up with their own keys so that even the designer of the cryptosystem cannot break it without knowing which key has actually been used.

For some applications (mostly military, diplomatic and covert actions), there is no reason for the designer of a general cryptosystem to publicly disclose the nature of his algorithms. Some additional safety can be obtained by keeping this information confidential. It is however crucial not to rely on this secrecy, for one never knows when it may be compromised. For this reason, reliability analyses of such systems should *always* be carried out under the assumption that the potential enemy knows all about the system, except for the actual key being used. And if the enemy in reality does not have this knowledge, so much the better. For other types of applications, such as large scale financial ones, it is in fact better to disclose how the cryptosystem works. Otherwise, users will always suspect the possible existence of a secret method to break the system.

An obvious requirement for the security of a general cryptosystem is a very large number of possible keys, so as to discourage *exhaustive search* (trying to systematically decipher a given ciphertext using each possible key until meaningful cleartext emerges). For instance, one might naively consider Caesar's cipher as an instance (with key $k = 3$) of the "general" cryptosystem consisting of replacing each letter in the plaintext with the kth following letter in the alphabet, where k is the secret key. This generalization is worthless because it accommodates only 25 non-trivial keys, making exhaustive search easy for anyone who suspects the nature of the encipherment (at least if the enciphered message has enough redundancy to allow only one meaningful decryption).

One should be aware, however, that there is no safety in large numbers alone. For instance, another generalization of Caesar's cipher consists of choosing as key an arbitrary permutation of the 26 letters of the alphabet, such as EROX...WM, and enciphering each plaintext letter according to this permutation (A → E, B → R, . . . , Z → M) so that BAD DAY becomes REX XEW. Considering that there are 26! different permutations of the 26 letters, which is more than 4×10^{26}, one might feel that exhaustive search on the key space is not feasible: it would take over ten billion years to try each possible key at the rate of one billion keys every second! Nonetheless, this (*mono-alphabetic*) *simple substitution cipher* is rather easy to cryptanalyse, if only because of the variation in natural-language letter frequencies [154, 116, 188]. Much safer cryptosystems have been designed with a significantly smaller key space.

Coming back to the classification, a general cryptosystem is *secret-key* if some secret piece of information (the key) has to be agreed upon ahead of time between any two parties that wish to communicate through the cryptosystem. In our previous example, if A enciphers a message using key EROX...WM and sends the ciphertext to B, it had better be the case that B knows which key was used for the encipherment.

This need for secure *key distribution* was not an insuperable problem in the days when cryptography was for the few, although foresight was necessary to prevent prohibitive delays before secure communication could be established. Now that cryptography has gone public, however, it is unreasonable to set up a network in which each *pair* of potential users shares a secret key in advance, because the number of keys would grow quadratically with the number of users.

In 1976, Diffie and Hellman laid the ground for overcoming this difficulty by proposing the notion of *public-key cryptography* [100, 101]. A similar idea was independently discovered by Merkle [175]. This was soon to be followed by Rivest, Shamir and Adleman's first proposed practical implementation [205]. Secure communication over insecure channels between two *totally unacquainted* parties was at last possible.

The key observation that lead to public-key cryptography was that whoever enciphers a message does not need to be able to decipher it. In such systems, each user selects a *private key* from which she obtains a pair of algorithms. She makes one of them available to everyone as her *public enciphering algorithm*, whereas she keeps secret the other one, which is the corresponding deciphering algorithm. This allows even a complete stranger to use her public algorithm to encipher a message for her; yet only she can decipher it through the use of her private algorithm. It goes without saying that such systems can only be secure if it is infeasible to figure out a deciphering algorithm from the corresponding public enciphering algorithm.

More recently, Goldwasser and Micali have set forward the notion of *probabilistic encryption*, which is a very interesting variation on the theme of public-key cryptography [132, 133, 40]. When a message is enciphered with probabilistic encryption, it becomes essentially just as hard for a cryptanalyst to figure out *any* information on the message than it is for him to recover its *entire* contents. Moreover, there exists a probabilistic encryption scheme that is faster than the leading public-key encryption scheme proposed thus far (RSA) – see sections 4.4 and 4.6. These cryptosystems are called "probabilistic" because enciphering the same cleartext message several times under the same key can give rise to completely different ciphertexts.

Other different approaches to the key distribution problem have been proposed. For instance, Alpern and Schneider's *keyless cryptography* can be used effectively in a network that hides the origin (but not the contents) of messages [6, 249, 93]. Shamir's *identity based cryptosystem* removes the need for key distribution, but requires a trusted center to create private keys [211]. We shall not discuss these concepts here. Finally, Bennett and Brassard built on the work of Wiesner [237] to develop *quantum cryptography*, which proposes completely different foundations for cryptography and bases its claims of security on quantum physics rather than mathematics and computational complexity theory [23, 17, 18, 19, 20, 21, 22]. The final chapter of this book is devoted to quantum cryptography.

CHAPTER 3

SECRET-KEY SYSTEMS

3.1. Definitions and Levels of Attack

A *secret-key cryptosystem* consists of a *key space* K and, for each $k \in K$, of a *cleartext message space* M_k, a *ciphertext message space* C_k, and a pair of functions $E_k : M_k \to C_k$ and $D_k : C_k \to M_k$ such that $D_k(E_k(m)) = m$ for each plaintext message $m \in M_k$. The cryptosystem is *endomorphic* if $C_k = M_k$ for each k. Given any key k, it must be easy to obtain efficient algorithms for computing E_k and D_k. The cryptosystem is used as follows for the purpose of secure communications. If A and B expect that they might eventually have to communicate privately, they must initially agree on some secret key $k \in K$. Whenever A wishes to send a specific $m \in M_k$ to B, she uses the enciphering algorithm E_k to produce $c = E_k(m)$; she sends c to B over an insecure channel; and B uses algorithm D_k to recover $m = D_k(c)$. In many practical cryptosystems, both M_k and C_k are finite sets, often independent of the key k (such as the set of all eight-character strings). In this case, it could be that the actual message m is too long to be enciphered directly. If this occurs, m must be broken in pieces and E_k must be used several times. We discuss this situation in section 3.5.

The least one could ask of a secret-key cryptosystem is that it should be infeasible for a cryptanalyst to infer m (or, worse, k) from eavesdropping on $c = E_k(m)$. Even a cryptosystem immune to this threat may however be weak under other circumstances. Secret-key cryptography distinguishes three levels of cryptanalytic attack.

- *Ciphertext only attack*: the cryptanalyst is given $c_1 = E_k(m_1)$, $c_2 = E_k(m_2)$, . . . , $c_i = E_k(m_i)$, the encipherings of i distinct unknown cleartext messages under the same unknown key. He is to infer the key k or, lacking this ability, as many among m_1 , m_2 , . . . , m_i as possible.

- *Known plaintext attack*: the cryptanalyst is given $c_1, c_2, \ldots, c_i$ as above, but also the corresponding $m_1, m_2, \ldots, m_i$. He is to infer k or, lacking this ability, he is to infer m_{i+1} from some *new* ciphertext $c_{i+1} = E_k(m_{i+1})$ enciphered using the same key.
- *Chosen plaintext attack*: the cryptanalyst gets to choose plaintext messages $m_1, m_2, \ldots, m_i$ and he is given the corresponding $c_1 = E_k(m_1)$, $c_2 = E_k(m_2), \ldots, c_i = E_k(m_i)$. He is to infer k or, lacking this ability, he is to infer m_{i+1} from some *new* ciphertext $c_{i+1} = E_k(m_{i+1})$ enciphered using the same key. (There are real life situations in which such a powerful chosen plaintext attack can be mounted – such as "identification-friend-or-foe" systems [154].)

The difference in power between these three levels of attack is best explained through our previous example of the simple substitution cipher. When we said that it is easy to cryptanalyse, we had in mind: under a ciphertext only attack. Although this is true, it does require some work. It becomes utterly trivial to break, however, under a known plaintext attack as soon as the available cleartext messages have used at least once each letter in the alphabet (of course, all but one suffices). Patience is not even needed under a chosen plaintext attack: the key (that is, the secret alphabet permutation) yields immediately if the value of E_k(ABCD...WXY) is available.

3.2. Information Theory, One-Time Pad, and Unicity Distance

What do we mean by "it should be infeasible for a cryptanalyst to infer m"? Two words deserve further explanation: "infeasible" and "infer". This section is relevant mostly for ciphertext only attacks.

In the setting of Shannon's classic information theory, "infeasible" means "mathematically impossible, regardless of available resources". For instance, suppose you toss a fair coin and, before looking at the outcome, you ask a friend to randomly decide whether to leave it as is or to flip it over. From looking at the end result of this experiment, it is impossible to infer the original outcome of the coin toss. (We shall give quite a different meaning to the word "infeasible" when we discuss public-key systems in chapter 4.)

The exact meaning of "infer" is more difficult to make precise without introducing a substantial amount of information theory. For a formal mathematical treatment, consult [214, 215, 119]. The cryptanalyst's ultimate goal is of course to figure out exactly and with certainty the key k or at least the plaintext message m. He may be satisfied, however, to learn some *probabilistic* information about m. Assuming the plaintext message is in English, the cryptanalyst has *a priori* information about it even *before* looking at the ciphertext. For instance, he knows that "hello" is a *more probable* heading than "xykph". The purpose of cryptanalysis is to increase this *a priori* information by modifying the probabilities associated with each possible plaintext message, thus making the correct plaintext more probable, although not necessarily certain.

Consider a situation in which the cryptanalyst has intercepted ciphertext "xtjja" and he knows (or suspects) it was enciphered using a simple substitution cipher. This tells him that the plaintext message has five letters, the third and fourth of which being the same and the others being distinct. He cannot conclude that the plaintext is "hello" because it could also be "teddy", for instance. Nonetheless, the *a posteriori* probabilities for these plaintexts increase relative to their *a priori* probabilities. He also knows with certainty (assuming he is correct about the nature of the cryptosystem) that the plaintext cannot be "peace" or "rambo", and the *a posteriori* probability for these plaintexts drops down to zero, regardless of their *a priori* probabilities.

Shannon defines a cryptosystem to achieve *perfect secrecy* if knowledge of the ciphertext yields no information whatsoever on the corresponding plaintext, with the possible exception of its length. In other words, the *a posteriori* probabilities after seeing the ciphertext are *exactly* the same as were the *a priori* probabilities. Such systems do indeed exist [234] and one might wonder why they are not the ultimate solution to all cryptographic needs. There are three main reasons why this is not so. As any secret-key system, they pose the problem of key-distribution. This difficulty is amplified by a theorem of Shannon's to the effect that perfect secrecy is *only* possible if the key space is at least as large as the cleartext message space, which amounts to saying that the secret key must be at least as long as the message itself *and* that the same