1

INTRODUCTION

This book is designed for use as a university text for year three, four, or honors level students. It is intended as a first approach to public key cryptography—no background in cryptography is needed. However, a basic understanding of discrete mathematics and algorithms and of the concept of computational complexity is assumed.

The major public key systems are presented in detail, both from the point of view of their design and their levels of security. Since all are based on a computationally difficult mathematical problem, the mathematics needed to construct and to analyze them is developed as needed along the way.

Each concept presented in the book comes with examples and problems, some of which can be done with limited computational capacity (a calculator for example) and some of which need major computational resources such as a mathematics-based software package or some independently written algorithms. Mathematica, Matlab, Magma [64], and Maple [65] are examples of packaged software that can be used easily to perform the necessary computations. For those who prefer open source software, see [28] where Sage is used for algorithms and examples. The book can be used without additional software resources by avoiding those problems which require them.

The software used by the author for the computationally expensive examples in this book was Maple. The solutions are presented with sufficient detail to permit an

Public Key Cryptography: Applications and Attacks, First Edition. Lynn Margaret Batten.

easy translation to any other language or package. Full solutions are given to all odd-numbered problems. For those wishing to use the book at a Master's level, an emphasis on the computational complexity of the cryptographic systems and or the attacks on them would provide a solid basis for a good course including programme writing. *Emphasis on the computational complexity of attacks on public key systems provides the user with a feel for the level of security provided.*

1.1 THE MEANING OF THE WORD CRYPTOGRAPHY

In this preliminary chapter, we present some of the history of cryptography and the reasons for the development of the systems that we see in use today. There are no exercises associated with this chapter, but the interested reader can follow up any of the references and links provided.

The words *"cryptography," "cryptology,"* and *"cryptanalysis"* are commonly interchanged. However, each of them has a slightly different meaning. The common beginning "crytp" comes from the Greek $\kappa\rho\upsilon\mu\mu\varepsilon\nu o\zeta$ or *kruptos* for "hidden." The ending "graphy" refers to writing and so the first word in the list means "hidden writing" and generally refers to the encryption part of establishing a system for transmitting secrets. We call such an encrypted string a "cipher" or "ciphertext." Normally, when a cipher is constructed, the idea is that there will be some person or persons who can "legitimately" decipher it and so find the hidden text. In order to legitimately decipher, it is understood that a person will hold what is referred to as a "key," a means of simply and efficiently determining the original text. On the other hand, without this key, it should not be simple to deduce the hidden text.

The last word, cryptanalysis, refers to an analysis of hidden things, or ciphers, to expose what is hidden; this word generally refers to the decryption or discovery component of the system when the analyst does not have a legitimate key with which to read a cipher.

Finally, the word cryptology is made up of the two components "hidden" and "study" and refers to the study of hidden writings or secrets. This word encompasses both the establishment of encryption methods and the analysis of a cipher in order to break it without the associated key. While "cryptology" would be the correct word for a discussion including both encryption techniques and analysis of these techniques with the intent of breaking them, many people use the word "cryptography" instead.

In the next section, we cover very briefly the introduction of, and changes to, symmetric key cryptography over thousands of years. This is followed by a brief introduction to public key cryptography. Recent applications of cryptography, in addition to simply hiding data, are mentioned in Section 1.5. Section 1.6 mentions current standards in the area of cryptography and their impact.

1.2 SYMMETRIC KEY CRYPTOGRAPHY

The hiding of secrets in written and pictorial form with the intent of passing on a message to a select few has been documented over thousands of years, going far back in time to

ancient Egypt [2, 36]. In many cases, it was used as a game so that the select few were able to have access to information not available to those excluded from the inner circle. However, it was also used in times of political tension and war to communicate securely, guarding secret information from the enemy.

Symmetric key systems are cryptographic systems in which decrypting is a simple method of reversing the encryption used. For example, if a message written in English is encrypted by replacing each letter with the one five places ahead in the alphabet (a is replaced by f, b by g, and so on), then to decrypt, the letters are simply moved five places back. A message written as a binary string may be encrypted by adding it to another, fixed, binary string. To decrypt, adding the fixed binary string again will produce the original message. Thus, to use a symmetric key cryptographic scheme, both the sender and the receiver use essentially the same key.

The simplicity of using the same key both to encrypt and to decrypt is off set by the difficulty of ensuring that all parties have the needed keys in a tense situation, and also when people may be widely dispersed geographically. In time of war, keys have to be physically delivered to personnel even in the remotest and most dangerous locations. In the late 1800s, the idea of a "code book" which listed which keys to be used on which dates was born. Both the transmitter and the receiver needed a copy of the same code book for this to work, but several months of communications could be based on the delivery of a single code book. (Serious users of encryption recognized the need for constantly changing the key!)

1.2.1 Impact of Technology

Despite its history of about 4000 years, cryptography only came of age in the 1800s with the invention of technologies such as the telegraph (for rapid communication over great distances) and manual rotary machines, followed in the early 1900s by electrical rotary machines [2]. David Khan, in his book *The Code Breakers* [22] explains that the electro mechanical rotary machine for cryptographic purposes was invented almost simultaneously around 1917–1919 by four different people in four different countries. None of these people became rich. One of them, the Swede *Arvid Damm*, died in 1927 and his company was taken over by another Swede, *Boris Hagelin* (1892–1983). Despite Hagelin's death, the company, Crypto AG (`http://www.crypto.ch/`), still operates in Zug, Switzerland. Figure 1.1 shows a machine sold by the company.

1.2.2 Confusion and Diffusion

As cryptography became less of an art form and more of a science in the 1900s, it was inevitable that at some point, someone would try to formalize the principal aims of a cryptographic system. Claude Shannon was one of the first to do so [48]. He argued that a cryptosystem designer should assume that the system may be attacked by someone who has access to it, as was indeed the case during the two world wars when machines were stolen and reverse engineered. He argued that the only point of secrecy should be the key, but that the system design should assist the security by incorporating "confusion" and "diffusion." "Confusion is intended to make the relationship between the key and

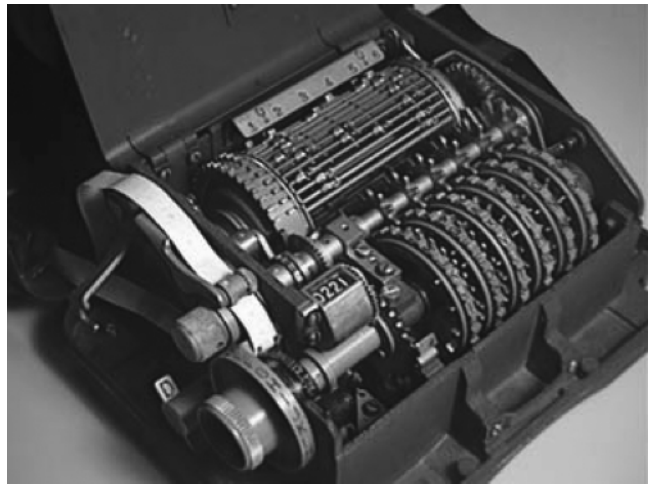

Figure 1.1. The M-209 encryption machine sold by Hagelin (from Wikipedia).

the ciphertext as complex as possible. Diffusion refers to rearranging or spreading out the bits in the message so that any redundancy in the plaintext is spread out over the ciphertext" [29]. In most symmetric key cryptosystems, confusion is provided by means of a *substitution* of some letters or symbols for others, whereas diffusion is provided by a *permutation* of the letters or symbols.

1.2.3 DES and AES

Horst Feistel (1919–1990) is believed to have been the first person to use the idea of a cipher with input broken into two parts of equal size, and iterated through several rounds in which functions and keys are applied, and right and left sides interchanged [29]. The concept is the basis for many symmetric ciphers in use today including the Data Encryption Standard (DES).

DES was the first commercially driven cryptographic product in history. By the mid-1900s, the effectiveness of cryptography for use by companies wishing to communicate in private led the U.S. government to work with IBM to develop the first fully specified cryptographic system on the open market. In 1976, the U.S. National Bureau of Standards declared it an official Federal Information Processing Standard.

In 1997, it was decided that the parameters of DES were now too small to provide the kind of protection needed, and a public, world-wide call for submissions for a new cryptosystem standard was made by the U.S. National Institute of Standards and Technology (NIST). After several years of analysis of submissions, much done by academics around the world, in 2002 a new standard, known as AES (for Advanced Encryption Standard), was chosen by NIST. While the AES does not incorporate a Feistel-type structure, such

as DES, it uses substitutions and permutations along with several rounds. More details on both DES and AES can be found in [52].

Further Reading. F.L. Bauer [2]; D. Khan [22]; S. Pincock [36]; C.E. Shannon [48].

1.3 PUBLIC KEY (ASYMMETRIC) CRYPTOGRAPHY

The one major problem that held back a general uptake of cryptography for use in business circles was that of exchanging keys. While for many years, governments had established methods of managing keys, business people were not interested in employing circumspect, and perhaps even dangerous, methods of exchanging keys. In the 1960s, this became known as the "key management" problem and it was to be another decade before a viable solution was found.

1.3.1 Diffie–Hellman Key Exchange

In 1976, Whit Diffie and Martin Hellman published a paper [13] describing a method of establishing a common key in a secure manner over an insecure channel. The method is based on exponentiation and the fact that exponents can be multiplied in any order with the same result. The method is described in detail in Section 2.3. However, this scheme was useful only for establishing keys and did not actually encrypt data. The search was still on for an encryption scheme that allowed anyone to send an enciphered message to any other person, without pre-establishing keys, such that only the targeted recipient could decrypt the message.

In retrospect, the solution is amazingly simple and the first example appears to have been developed independently from two sources.

Basically, the idea is for each person to have two keys, one to encrypt and one to decrypt. The two keys would have to be bound together in some fundamental way in order for them to "invert" each other, but it should be impossible for an attacker to derive one from the other. The encryption key would be published, as in a telephone book. Only the recipient would know his/her decryption key; it would not be revealed to anyone else. This idea entirely solved the problem of exchanging keys, except for the fact that, initially, no one actually had a real way of setting up such a scheme.

1.3.2 RSA

In 1978, the first actual method for implementing such a scheme was published by Ron Rivest et al. [46] and is now widely known by the first letter of each of the authors' names as RSA. An RSA patent was filed in the United States on December 14, 1977, and approved as #4,405,829 titled "Cryptographic Communications System and Method" to the Massachusett's Institute of Technology, Rivest, Shamir, and Adelman. However, since the work had been published before the patent application, it could not be patented

under European and Japanese law. (The RSA United States patent expired in 2000.) The company, RSA Data Security, was formed shortly thereafter and was granted an exclusive license on the RSA patent. In 2006, EMC Corporation (`www.emc.com`), a global information management and storage company, bought RSA which continues to operate as EMC's security division.

RSA security was based on the difficulty of factoring large numbers. At the time the company was established, the state-of-the-art research in factoring was not fully understood. To gauge what was known in this area, RSA Data Security put out the *RSA Factoring Challenge* in 1991 to encourage research into computational number theory and the practical difficulty of factoring large integers and the breaking of RSA keys used in cryptography. We focus on this in Chapters 7 and 8. A cash prize was offered for the successful factorization of some of the numbers posted. The smallest of them, a 100 decimal digit number called RSA-100, was factored by April 1, 1991, for a US$1000 prize. The RSA challenges ended in 2007.

In 1997, it was finally revealed that members of the British intelligence agency Government Communications Headquarters (GCHQ) had also invented essentially the same scheme early in the 1970s. See Steven Levy's book [26] for the interesting story of the parallel development.

1.3.3 ElGamal

The ElGamal cryptographic algorithm was invented a few years after the RSA scheme, developing from the PhD thesis of Taher ElGamal, which was awarded in 1984. The underlying idea on which the security is based is quite different from that of RSA. In ElGamal, the target is to determine the exponent in an equation of the form $a = b^x$, where a and b are known. The inventor did not apply for a patent on his scheme.

All known public key schemes are far more computationally intensive than symmetric key schemes. For example, a disadvantage of the ElGamal system is that the encrypted message becomes very big, about twice the size of the original message. Similarly, RSA is slower than DES by a factor of about 1000. For this reason, public key schemes are traditionally used only for small messages such as secret keys, whereas symmetric key schemes are retained for sending large messages.

Whether using a symmetric key or public key approach to encryption, the underlying mathematical formulation needs to be based on a finite system in order to ensure that infinite loops are avoided in computations. The arithmetic of such systems is developed in Chapter 2. Specifically, congruence arithmetic underpins all known cryptographic systems.

A second common feature of symmetric and asymmetric encryption is the use of both an encryption and a decryption key where data is transmitted over an insecure channel. This is illustrated in Figure 1.2.

While many public key cryptosystems have been proposed, only a few have withstood the test of time to remain in use today. In this book, we cover in detail three of those systems that have endured. RSA and ElGamal have been mentioned here. The third system we consider is based on elliptic curves and presented in Chapter 5. As mentioned

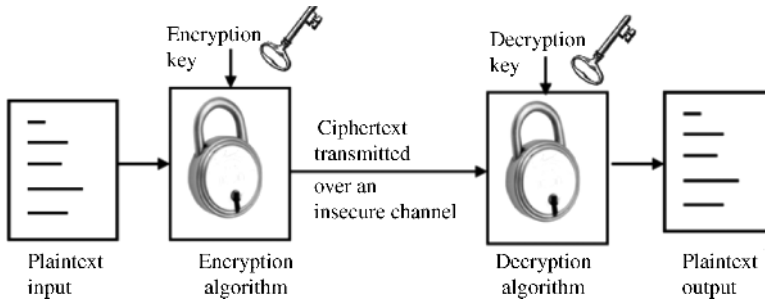

Figure 1.2. Transmitting encrypted data over an insecure channel.

in Section 5.2, the elliptic curve system is very efficient and so useful to implement on small devices.

Further Reading. W. Diffie and M. Hellman [13]; R. Rivest, A. Shamir and L. Adlemen [46]; S. Levy [26]; T. ElGamal [14].

1.4 KEY ESTABLISHMENT

The previous two sections referred to the importance of keys in cryptosystems but did not directly address the question of how to establish them. In this section, we shall only mention answers to this question that relate to public key cryptography.

In theory, anyone wishing to encrypt data can produce their own public/private key pair based on any public key system, but in practice, people rely on a third party to provide them with keys. Out sourcing of key generation has the following benefits:

- the third party does the expensive computations;
- the third party specializes in key generation and knows the current best methods of securely and efficiently generating them;
- the third party takes on the liability for any problems arising from key failure.

However, it also has disadvantages:

- the third party knows your private key and so must be completely trust-worthy;
- the third party may be taken over by an organization which you do not trust.

However public/private key pairs are generated, trying to set up keys between two people over an insecure channel has associated problems. See the description of the "Intruder-in-the-Middle" attack in Section 6.3. (This attack is more commonly known as the "Man-in-the-Middle" attack and can be further studied in many other books on

cryptography.) The concept of a "certificate" to tie a user to the key and a "certificate authority" to issue such certificates was born from trying to stop this attack. Section 13.3 of [28] gives a better discussion of these concepts.

Further Reading. The book *Protocols for Key Establishment and Authentication* by Boyd and Mathuria is a good source of further reading on this topic [4].

1.5 CRYPTOGRAPHY—MORE THAN JUST HIDING SECRETS

Along with the move from the government to the corporate sector, and the ease of use resulting from the pervasiveness of computers, a number of applications of cryptographic techniques, other than simply that for hiding data, have been developed. These have been motivated by needs of the digital age: how to confirm that a message received indeed came from the sender purported (it is quite easy to change a sender name in most e-mail systems), how to prevent a sender from claiming that they did not in fact send the message you received, and how to ensure that the message received was the one sent and had not been altered.

The most significant recent applications of cryptography are therefore for identification of senders, authentication of senders and recipients as well as the messages themselves, and digital signatures applied to messages.

1.5.1 Digital Signatures

A digital signature is a method of applying data to a message which identifies the sender of the message in the same way that a written signature on a piece of paper confirms authorship. All known public key systems have features that allow for this possibility. Digital signing is the basis of a number of the other applications mentioned earlier Chapter 6 is dedicated to this topic.

In using a public key scheme to send a signed message to Bob, Alice can simply apply her secret key to the message, essentially "encrypting" it with her decryption key. when Bob receives the message, he can verify that it was "signed" by Alice by applying Alice's public (encryption) key to it (see Figure 1.3). The result should be a readable message that makes sense to Bob. If it was not signed with Alice's private key, then

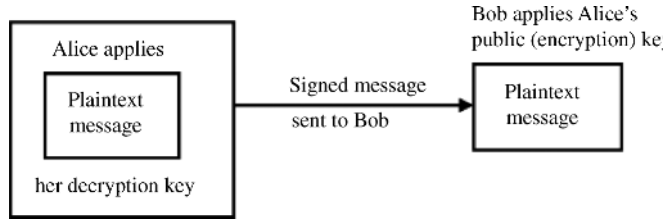

Figure 1.3. Alice signs a message for Bob.

applying her public key to it would result in nonsense, so Bob can be sure that it was signed by Alice. We discuss these procedures in more detail in Chapter 6.

1.5.2 Authentication

Parties entering into a communication over a digital pathway need to be able to identify each other. In addition, the data they transfer to each other needs to be authenticated as to origin, content, time sent, and so on. These issues all relate to an "intruder-in-the-middle" attack described in Section 6.3, where an individual places herself between two unsuspecting correspondents and reads, alters, and then passes on data between them. The integrity of data is preserved when such authentication is available.

1.5.3 Nonrepudiation

This ensures that a computer user cannot deny any actions or commitments made previously. If a dispute arises, a trusted third party can be brought in to assess the earlier communications and determine if the cryptographic protocols were in place to provide nonrepudiation.

1.6 STANDARDS

Many public key-based protocols have now become entrenched in standards. While this is good for those chosen, it means that it is difficult for any bright new idea to gain a foothold in the market place. Who chooses these standards? It is usually government or government-sponsored organizations that do so. One such organization is the National Institute of Standards and Technology, or NIST, a U.S. federal government agency that works with industry to develop and apply measurements and standards. For instance, NIST has established a standard for digital signatures, which can be found at `http://www.itl.nist.gov/fipspubs/fip186.htm`. We discuss this standard in Section 6.2. NIST is also the organization which established U.S. government standards for symmetric key and public key cryptography and hash functions (this last being the subject of Section 6.1).

Other countries have similar organizations determining standards. In Australia, the information security division of the Defence Signals Directorate determines cryptographic standards (see `http://www.dsd.gov.au/infosec/evaluation_services/epl/epl.html`). In Canada, the Standards Council of Canada is charged with recommendations on the choice of cryptographic protocol. See `http://www.scc.ca`. In the United Kingdom, it is the National Technical Authority for Information Assurance that provides this service; they can even point to products. See `http://www.cesg.gov.uk`. In the People's Republic of China, the Ministry of Public Security is responsible for protection of computer systems and the choice of cryptographic protocol to employ.

There is no international standard; each country makes its own choices. However, the Organization for Economic Co-operation and Development (OECD) issues guidelines for the use of information security systems that are often adopted by those

governments or organizations without the resources to develop them independently. See http://www.oecd.org/document/42/0,2340,en_2649_34255_15582250_1_1_1_1,00.html.

1.7 ATTACKS

On the one hand, many people and organizations wish to be able to communicate information over insecure channels in such a way that only targeted receivers can read it; on the other hand, there are many individuals and organizations who wish to gain access to information which they would not normally receive. Examples include governments allied in war sending each other plans to combat other governments, the head office of a large corporation distributing the plans for the next version of their product to national offices, individuals communicating instructions on transfers of money to their banks.

In each of these cases, the communicating parties might use encryption to make sure that an interceptor cannot read the information. In each case, there are people waiting for an opportunity to capture, read, and possibly change the information being transmitted.

Whenever methods are implemented to secure data, care must be taken to do so without leaving weak components which are vulnerable to attackers. In this book, for each method of encryption we introduce, we also consider attacks against it. However, we restrict ourselves to the attacks against the underlying mathematical construction of the encryption. There are countless other ways of attacking any security system ranging from bribing or deceiving a person who is responsible for it to physically destroying the entire communication channel with a bomb blast. Unfortunately, the attacker has the advantage of a multitude of possible attacks many of which are not foreseen by those who construct the encryption system.

The public key cryptosystems considered in this book are based on only two fundamental mathematical concepts known as integer factoring (see Section 4.1) and the discrete logarithm (see Section 2.3). In Section 2.4, we consider attacks against the discrete logarithm which jeopardize key establishment and the ElGamal (Section 3.3) and elliptic curve (Section 5.2) cryptosystems. In Section 4.3 and Chapters 7 and 8, we consider attacks based on integer factoring that affect the RSA cryptosystem. The problem of factoring integers has been of interest to mathematicians for thousands of years and so there is an enormous body of mathematical literature here. Thus, the techniques for attacks based on this are now very sophisticated.

While there are many other public key cryptosystems in the literature, many of them are still based on one of the two concepts mentioned earlier and so vulnerable to attacks on discrete logarithms and factoring. We provide some references in the further reading section below.

Further Reading. Chapter 8 of the *Handbook of Applied Cryptography* [29] discusses several public key cryptosystems. Like RSA, the Rabin scheme is secure only if integer factoring cannot be done. The McEliece scheme mentioned there (Section 8.5)

has security based on the problem of decoding linear codes and the Merkle–Hellman scheme security is based on subset sums (Section 8.6); while the first of these schemes is still believed to be secure, the second has been shown to be insecure but stronger versions are known. (See Note 8.41 of [29].) The Cramer–Shoup scheme, like ElGamal, is based on the discrete logarithm. A description of the scheme and a proof of security can be found in [27]. The further reading sections in subsequent chapters also point the reader to more work on attacks.