Introduction to Cryptography

Mark Vandenwauver

Katholieke Universiteit Leuven, *Laboratorium ESAT-Groep COSIC*

Contents

Introduction

The origin of the word cryptology lies in ancient Greek. The word cryptology is made up of two components: "kryptos", which means hidden and "logos" which means word. Cryptology is as old as writing itself, and has been used for thousands of years to safeguard military and diplomatic communications. For example, the famous Roman emperor Julius Caesar used a cipher to protect the messages to his troops. Within the field of cryptology one can see two separate divisions: cryptography and cryptanalysis. The cryptographer seeks methods to ensure the safety and security of conversations while the cryptanalyst tries to undo the former's work by breaking his systems.

The main goals of modern cryptography can be seen as: user authentication, data authentication (data

integrity and data origin authentication), non-repudiation of origin, and data confidentiality. In the following section we will elaborate more on these services. Subsequently we will explain how these services can be realized using cryptographic primitives.

Cryptographic services

User Authentication

If you log to a computer system there must (or at least should) be some way that you can convince it of your identity. Once it knows your identity, it can verify whether you are entitled to enter the system. The same principal applies when one person tries to communicate with another: as a first step you want to verify that you are communicating with the right person. Therefore there must be some way in which you can prove your identity. This process is called user authentication. There are several ways to obtain user authentication.

You can give him something only you can know: a password, a (predesigned) user-id, a pincode, and so on. Or you could have some specific items with which you can identify yourself: a magnetic strip card, a smart card (a hand-held computer the size of a credit-card), a token. One might make use of biometric properties; it is a well-known fact that fingerprints, the shape of the hand and retinal pattern of a person are good decision criteria. These however require specialized equipment and thus a big investment. However, these biometric systems are not perfect: some legitimate users will inevitably fail the identification and some intruders will be accepted as genuine. Other techniques include measurements of how a person types his name or writes his signature, or can take into account the location of the user.

Figure 1: User authentication

For the time being the first two methods are the ones generally applied, and many practical systems use a combination of both. Since the user's memory is limited, this information should not vary too much over time. Whether it is a password, a pincode or a user-id, all these items are being defined at a certain time and often don't change from there on. One might argue that you could change your password, but this is not done each time you access the computer. This indicates that someone who can eavesdrop this information, will later be able to impersonate the user. A similar observation holds true for a magnetic strip card or memory chip. All these systems provide static authentication only.

If the user possesses a device which can perform simple computations, the security can be increased significantly by introducing the well-known challenge-response idea. If a person tries to identify himself to the system, the system generates a random challenge and sends it to the person or to his device. In case of a token (a mini-calculator), the user will have to enter the challenge on the keyboard. The device will then compute the corresponding response, using secret information which has been assigned to him. This response is then sent back to the system, which verifies it (see figure 1). If more sophisticated protocols are used, the verifier does not need secret information (this requires public-key protocols), or will even not learn the secret of the users (this requires zero-knowledge protocols). Note that in this case the procedure does not authenticate the user but rather his device. In order to increase the security, the user should authenticate himself with respect to the device, using something he alone knows. This makes the device useless if it is stolen.

In general, one also requires that the computer authenticates itself to the person logging on. If both parties are authenticated to each other, we use the term *mutual authentication*.

Data authentication

Data authentication consists of two components: the fact that data has not been modified (data integrity) and the fact that you know who the sender is (data origin authentication).

Data integrity

A data integrity service guarantees that the content of the message, that was sent, has not been tampered with. Data integrity by itself is not meaningful: it does not help you to know that the data you have received has not been modified, unless you know it has been sent directly to you by the right person. Therefore it should always be combined with data origin authentication.

You should always be alert for possible intruders in your network or in your communication system. A well-known example is the Internet that connects universities and companies world-wide. Electronic mail over the Internet does not offer any security. As a consequence, an educated computer user can tap into the messages that are being transmitted over the line. It is very easy to read and modify someones electronic mail, which is commonly seen as being private.

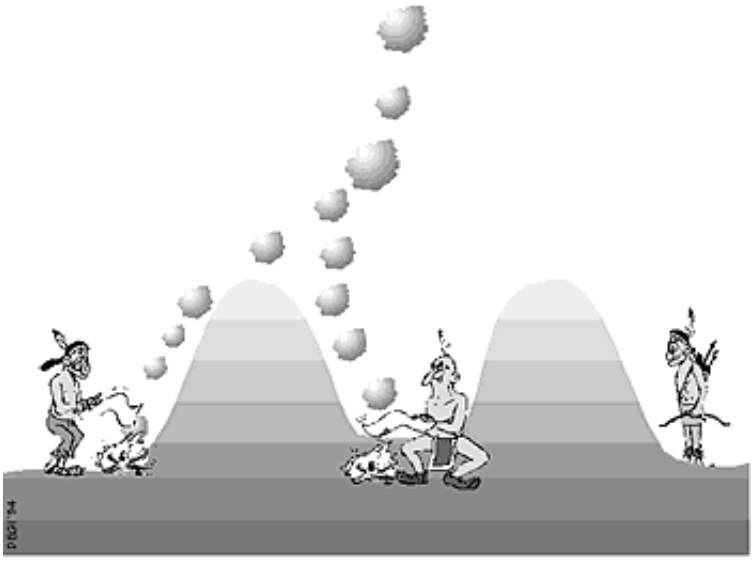

Figure 2: Data integrity

In general, we take the point of view of figure 2. We have A(lice) who sends a message to B(ob). There is also an enemy who taps the line between them. If you don't support data integrity, this enemy can just change the message and then relay it to B. B will not see that the message has been tampered with and will assume A really intended it the way he got it. One could argue that active wire-tapping is difficult. In general wire-tapping is only a matter of cost: tapping a telephone line is obviously easier than tapping a coaxial cable or a micro-wave. Active wire-taps (modifying and then relaying the messages) are also more difficult than passive wire-taps (listening in on the messages).

Data origin authentication

Here one wants to make sure that the person who is claiming to be the sender of the message really is the one from whom it originates. In figure 3, if A sends a message to B, but the enemy intercepts it and sends it to B, claiming A has sent it, how can B be sure of the real origin of this data? A variation on this theme is: the enemy could send a message to B claiming it A is the originator. Thanks to cryptography, there are techniques to ensure against this type of fraud.

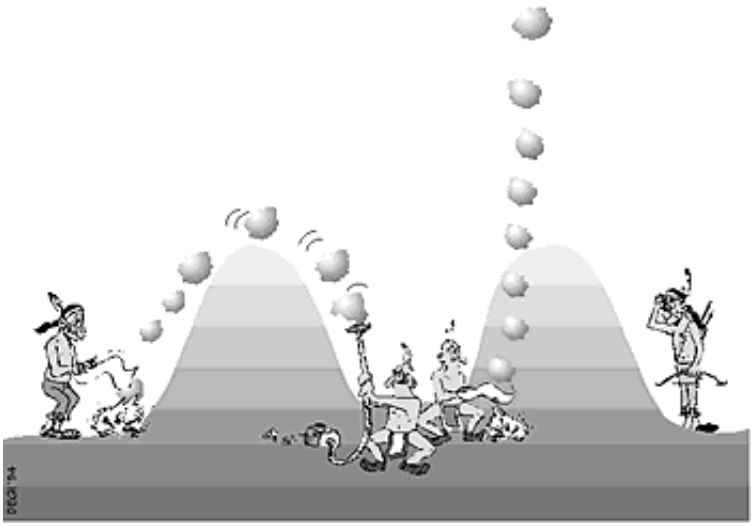

Figure 3: Data origin authentication

Non-repudiation of origin

Non-repudiation protects against denial by one of the entities involved in a communication of having participated in all or part of the communication. Non-repudiation with proof of origin protects against any attempts by the sender to repudiate having sent a message, while non-repudiation with proof of delivery protects against any attempt by the recipient to deny, falsely, having received a message.

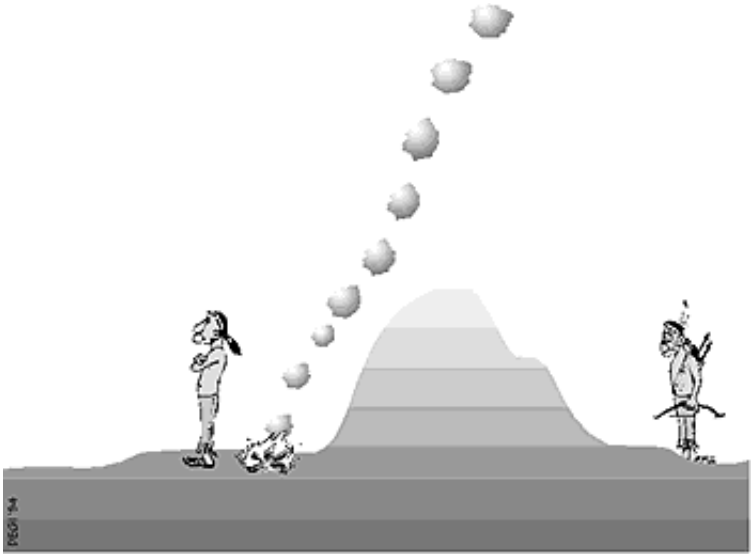

Figure 4: Non-repudiation of origin

An example will illustrate the importance of non-repudiation of origin. Suppose B is the owner of a mail-order company and he decides to let his customers order through electronic mail. For him it is really important that he can show to an arbitrary third party that A really ordered the things he is claiming otherwise it would be easy for a customer to deny the purchase of the goods (see figure 4). In a paper and pencil world, non-repudiation is provided by a manual signature.

Data confidentiality

This aspect of data security certainly is the oldest and best known. The example of Caesars cipher given in the introduction clearly demonstrates this. The fact that confidentiality was considered to be much more important than authentication of both sender and data, together with non-repudiation of origin can be explained as follows: the latter services have been provided implicitly by the physical properties of the channel: a letter was written in a recognizable handwriting, with a seal and a signature.

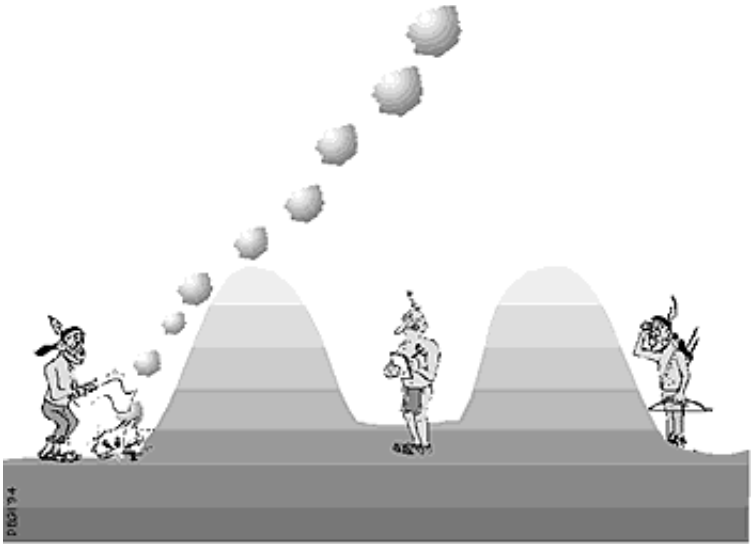

Figure 5: Data confidentiality

With data confidentiality we try to protect ourselves against unauthorized disclosure of the message. Referring to figure 5, if A sends a message to B, but the enemy intercepts it, one wants to make sure that this enemy never understands his contents. Confidentiality protection is very important in the medical world and also in the banking sector. World-wide there are several million transactions each day and all of these have to be passed from one financial institution to another. If there were no way to protect confidentiality, everybody would be able to see who had purchased what, who has made what kind of withdrawal, and so on.

Clearly this would violate individuals and companies rights to privacy. In order to provide confidentiality, it is necessary to transform the message with a cipher.

Cryptographic primitives

The above cryptographic services can be realized by several cryptographic primitives: we distinguish between primitives for encryption, primitives for authentication, and cryptographic protocols. Encryption primitives can be used to provide confidentiality, authentication primitives can be used to provide data authentication. We will also discuss protocols for user authentication and for key management.

Figure 6: Encryption

Encryption primitives

In cryptography one often makes use of encryption. With encryption we transform the cleartext (or plaintext) into ciphertext. To get back to the original text, we apply the inverse transformation, called decryption. These transformations themselves are public: this makes it possible to analyze these algorithms and to develop efficient implementations. However they use a secret parameter : the keys which are known only by the sender and/or the receiver.

This key is the only thing one needs to know in order to encipher or decipher. Thus it is really important to manage one's keys and keep them secret where necessary.

We discuss two types of encryption primitives, symmetric or conventional ciphers and asymmetric or public-key ciphers.

Symmetric ciphers

Basically there are two kinds of encryption-schemes. The oldest ones and most used until now are the symmetric ciphers. In these schemes, the key used to decipher the ciphertext is equal to the one used to encipher the plaintext.

The best known cipher in this category is the Data Encryption Standard (DES), that was adopted in 1977 by the American NBS (National Bureau of Standards) as FIPS 46. Since then it has been used all over the world and until now no major flaws have been discovered.

The DES makes use of a 56-bit key which is unfortunately short. Researchers have estimated that exhaustively searching for all possible values of this key in 1 day will currently require an investment of about US$ 200,000 (this requires only a few pairs of plaintext and corrsponding ciphertext).

During the last years E. Biham and A. Shamir, and later M. Matsui have published attacks which break DES in the academic sense (i.e., they require significantly less operations), though this is no threat to the DES in practice, since they require huge amounts of known or chose plaintexts respectively.

Better security can be achieved using the triple-DES. In this way, we effectively obtain a key of 112 bits and this is sufficiently large. At the same time, one is protected against further improvements in academic attacks on the DES.

It is not sufficient to choose a secure cipher; one also has to specify a secure mode of operation. Depending on the nature of the communication channel or storage space, one will choose between Cipher-Block-Chaing (CBC), Cipher-Feedback (CFB), and Output-Feedback (OFB) (as specified in FIPS 81). Encryption block by block (or Electronic Code Book (ECB) mode) will only be used for encryption of keys.

Asymmetric ciphers

The asymmetric or public-key ciphers are the most recent cryptographic tools. In contrary to the symmetric systems the key used to encipher and the one used to decipher are different. Each partner thus has two keys.

He keeps one key secret and makes the other one public. If A wants to send a message to B, he just enciphers it with B's public key. Since B is the only one who has access to the secret key, B is the only one who can decipher the message and read the contents.

Figure 7: Key management with asymmetric cipher

The most popular public-key cipher is the RSA system (RSA stands for Rivest, Shamir and Adleman, the names of the three inventors). The security of this scheme is related to the mathematical problem of factorization: it is easy to generate two large primes and to multiply them, but given a large number that is the product of two primes, it requires a huge amount of computation to find the two prime factors.

At the moment of writing of this text, the biggest number that has been factorized was about 430 bits long and attacks on numbers of 512 bits have been announced for 1997. Therefore the absolute minimum length of the key in the RSA system has to be set at 640 bits; 768 or 1024 bits are required for any system that requires security for more than a few months.

Symmetric versus asymmetric ciphers

The biggest drawback of the asymmetric systems up until now has been the relative low performance compared to the symmetric ones. For example, the implementation of DES on a 586/90 PC could achieve a 15 Mbit/s encryption rate while the RSA implementation on the same PC has only a 6 Kbits/s rate. Thus as you can see the DES is typically 1000 times faster than the RSA-scheme.

Public-key systems provide significant benefits in terms of key management: if every user generates his own key, only an authentic channel is required, eliminating (expensive) secret channels like couriers.

Figure 8: Key management with asymmetric cipher

In systems without a central trusted server, the number of keys can be reduced. Indeed, suppose we have a network of n users each of whom wanting to communicate with the others. Since each communication requires a secret key, the total number of keys required equals n*(n-1)/2.

In the public-key system each user only needs a personal public/secret key pair, yielding a total of only 2n keys. If n equals 1000 this would mean 2000 versus 499500. In systems with a central management system, both approaches require the same number of keys. However, the central system can be off-line in the case of use of public key technology, which reduces the cost and minimizes the security risks.

In practice one thus often encounters hybrid systems in which one uses a public-key system for the distribution of the secret keys and a symmetric cipher for the bulk encryption of the data.

Authentication primitives

One-way functions and hash codes

A one-way function is defined as a function f such that for every x in the domain of f, f(x) is easy to compute; but for virtually all y in the range of f, it is computationally infeasible to find an x such that y=f(x). In addition one requires that it is hard to find a second pre-image: given an x and the corresponding value of f(x), it should be hard to find an x' different from x which has the same image under f.

One-way functions are used to protect passwords: one will store a one-way image of the password in the computer rather than the password itself. One applies then the one-way function to the input of the user and verifies whether the outcome agrees with the value stored in the table.

A hash function is a function which maps an input of arbitrary length into a fixed number of output bits. In

order to be useful for cryptographic applications, a hash function has to satisfy some additional requirements. One can distinguish two types of hash functions. A MAC (Message Authentication Code) that uses a secret key, and an MDC (Manipulation Detection Code) that works without a key. For a MAC one requires that it should be impossible to compute the MAC without knowledge of the secret key. For an MDC one requires that it is a one-way function, and - in most cases - that it is collision resistant, which means that it should be hard to find two arguments hashing to the same result.

Hash functions can be used to protect the authenticity of large quantities of data with a short secret key (MAC), or to protect the authenticity of a short string (MDC). Sometimes an MDC is used in combination with encryption, which can yield protection of both confidentiality and authenticity.

There are several schemes which have been proposed for use as hash functions. The widely used construction for a MAC is the CBC mode of the DES (with an additional output transformation), as specified in ISO-9797. Several MDC's have been constructed based on the DES. Other dedicated designs are SHA (Secure Hash Algorithm or FIPS 180), and RIPE-MD 160. These hash functions achieve a very high throughput (Mbit/s), even in software implementations.

Digital signature

Public-key techniques can also be used for other purposes than for enciphering information. If Alice adds some redundancy to her message and transforms the result using her secret key, anyone who knows Alice's public key can verify that this message was sent by Alice (by verifying the redundancy). In this way one can create a digital signature, which is the equivalent of the hand-written signature on a document.

Since it is not physically connected to the signed data or the originator, it will depend on this data and on the secret key of the originator. Several signature schemes have been proposed. The RSA public-key cryptosystem is the only one which can be used for both enciphering and digital signatures. Schemes which can only be used for digital signature purposes are the DSA and the Fiat-Shamir scheme.

Note that it is possible to produce a digital signature based on conventional ciphers like the DES. However, these schemes are less efficient in terms of memory and computations. Other constructions use a conventional cipher in combination with tamper resistant hardware: this offers only a limited protection.

Assume Bob has received from Alice a digitally signed message. If Alice subsequently denies having sent the message, Bob can go to a third party (e.g., a judge), who will be able to obtain Alice's public key. Subsequently he can verify the validity of the signature. In this way a digital signature can provide non-repudiation of origin. It is easy to see that it provides in addition data authentication, i.e., data integrity and data origin authentication.

Hash functions versus digital signatures

Hash functions can only be used in a situation where the parties mutually trust each other: they cannot be used to resolve a dispute (unless one uses, in addition tamper resistant hardware).

As in the case of encryption, hash functions tend to be three orders of magnitude faster than digital signatures. This explains why in general one will first compute the hashcode of the message with a fast hash function and subsequently apply the digital signature to this short hashcode. This provides digital signatures

which are not only faster and shorter, but also more secure.

Cryptographic protocols

A cryptographic protocol is an interaction between one or more entities to achieve a certain goal. In fact, encryption and digital signatures can be seen as a special case of cryptographic protocols.

While a huge number of protocols have been developed, we will restrict this section to two types of protocols: protocols for user authentication and protocols for key management.

User authentication protocols

The design of cryptographic protocols for user authentication is very complex. A large number of protocols have been presented in the available literature, many of which exhibit some weaknesses. The simplest protocol providing unilateral authentication consist of sending a password.

More complex challenge-response protocols can be designed in which the user does not transmit his secret information. They are based on an encryption algorithm, a MAC or a digital signature and the use, in addition, of so called nonces (never used more than once) : random numbers, sequence numbers or time stamps. More complex protocols are required to achieve mutual authentication.

Key Management Protocols

On e of the main links in the cryptographic keychain is the key management protocol: every cryptographic service will make use of cryptographic keying material, whose confidentiality and/or integrity has to be protected. For the distribution of this keying material, one can use a new cryptographic primitive, and ultimately, a physical channel.

In this way one builds a key hierachy: secret keys for bulk encryption with a symmetric cipher system will be encrypted using an asymmetric cipher system and signed with a digital signature scheme. The public keys of the asymmetric cipher can be distributed via an authentic channel which can be provided for example by combining conventional mail with voice authentication. An alternative is to sign these public keys with a single master key: now one only has to distribute a single master key via an authentic channel. These signed public keys are called certificates. The central authority certifies that a certain public key belongs to a particular user. The commonly used scheme nowadays in based on the ITU-T X.509 recommendation.

Note that there also exist public-key protocols which result in the agreement of a secret key between two parties, by exchanging public keys or parameters. A well known example in this class is the Diffie-Hellman key agreement scheme. This protocol is different from a key transport protocol, in which one party generates the secret key and enciphers it with the public key of the other party. The key agreement protocols have the advantage that they result in an increased security level.

In the context of public-key cryptography, revocation of public keys is very important: once the user's secret key is compromised, anybody can read his messages or forge his signatures. Although public-key systems require no on-line central management system, the system has to provide a means to protect the user in the case by warning the other users that his public key is no longer valid.