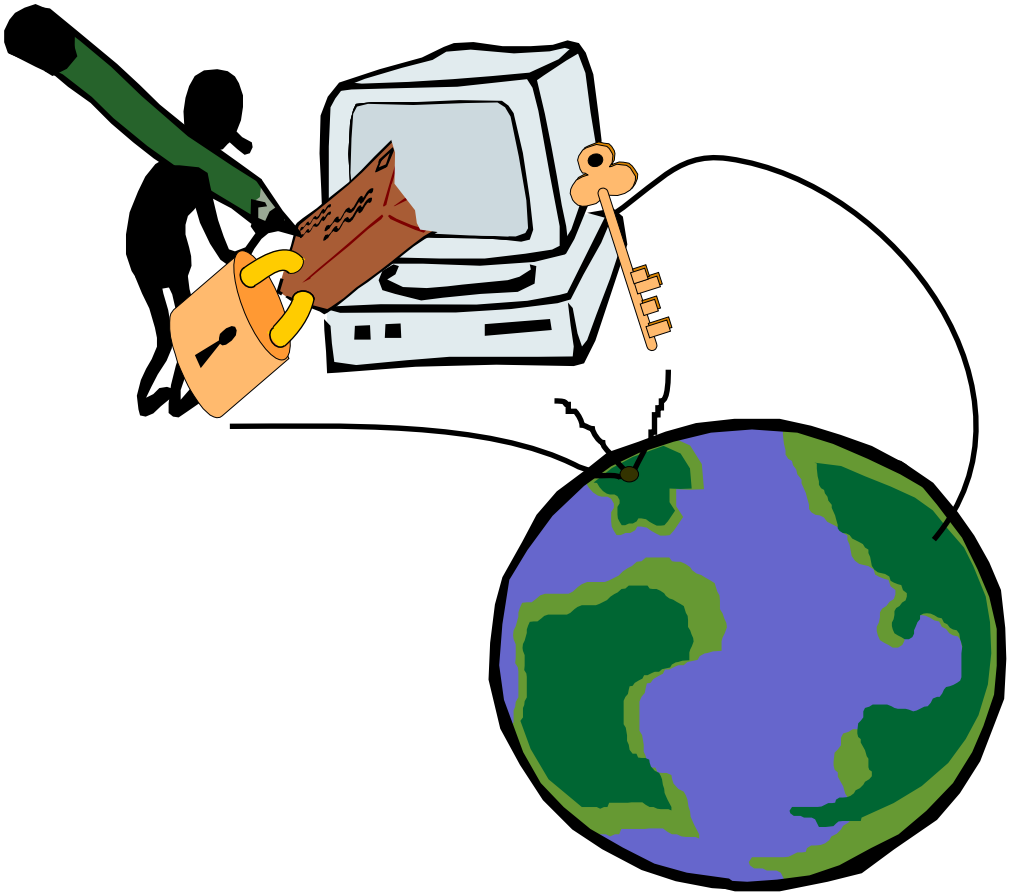
Digital Signatures and the Public Key Infrastructure



S.M. van den Broek

Department of Econometrics
Faculty of Economics
Erasmus University Rotterdam

Digital signatures and the Public Key infrastructure

S.M. van den Broek

Department of Econometrics
Faculty of Economics
Erasmus University Rotterdam

Rotterdam, 11th April 1999

scriptie begeleider
dr. ir. J. van den Berg

This thesis was typeset by the author with the LaTeX 2ε Documentation System

Acknowledgments

I am very grateful for the advice and support of my advisor, Jan van den Berg, for his sharpness of mind and his dedication to detail. Furthermore I would like to thank Deloitte & Touche Enterprise Risk Services, especially Xander Stox, for giving me the opportunity to explore one of the most interesting issues that are currently changing the conduct of business. I would also like to thank Wilco van Ginkel and Remco Kroes for reviewing my thesis and providing me with good feedback and Sanne Bakker for her understanding and support during the writing of this thesis.

Stefan van den Broek
Rotterdam, 11th April 1999

Contents

List of Figures

Chapter 1

Introduction

1.1 Information transfer

Historically communication has been between people talking directly with one another in person. As people became able to exchange ideas or information in writing, communication became possible over a greater distance. Letters could be written and sent by post. As technology evolved it became possible to exchange information verbally with the help of a direct cable connection or a telephone. These steps can be visualized with the help of figure 1.1.

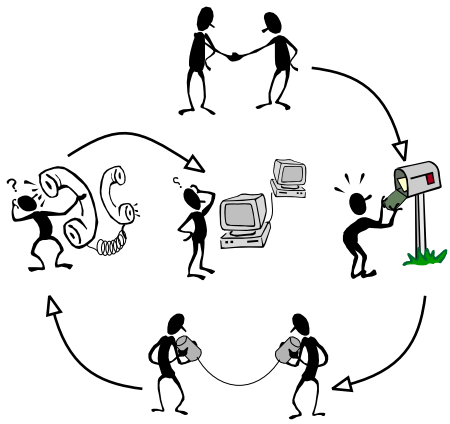


Figure 1.1: Communication media

More and more communication became possible with methods that have less and less features to identify with whom one is actually communicating. Until recently it was possible to recognize the sender of the information by handwriting and the signature at the bottom of a letter or by speech recognition. New technologies have made it possible to exchange information with the help of computers. This information

can be sent across a network[1]. The problem now arises that the information received over a network does not have any features that make it unique. Another problem concerns the sender not knowing the receiver and vice versa. Some kind of method is necessary to establish trust in one another.

1.2 Features of information transfer across a network

Basically the issue consists of transferring information from A to B as shown in figure 1.2.

What do we expect from this information transfer?

1. A network should be *available* in order to make any communication possible. At the same time the usage of a system should be restricted to those users that are authorized;

2. If B receives information then he[2] should be able to determine who sent it. At the same time the sender should be able to determine who receives the information. *Authentication* of the *sender* and the *receiver* is essential in order to judge the value of information exchanged;

3. If A sends a message to B then B should not be able to later deny having received this information. A should also not be able to deny having sent the information. If A or B could somehow dispute ever having taken part in the communication then this cannot lead to a trustworthy communication media. Thus *Non-repudiation of delivery and receipt* is essential;

4. It should not be possible to change the contents of the information or in case of this happening then any change should be made visible. Thus everyone receiving a message should be assured it is genuine. Another word for this is *content integrity*;

5. If private information is transferred then both A and B should be assured this cannot be read by anyone but A and B. This is called *confidentiality.*

In this thesis security will imply adhering to these features.

1.2.1 Controlling network security

> *"As long as networks are operated within the organization boundaries, the organization has the responsibility as well as the opportunity to operate the networks in conformance with the qualitative and quantitative demands and requirements that apply to the organization"* [RvdB98].

[1] With the term *network* is meant an interconnected collection of autonomous computers. Two computers are said to be interconnected if they are able to exchange information [Tan96].

[2] Whenever the word he is said this could also imply a she.

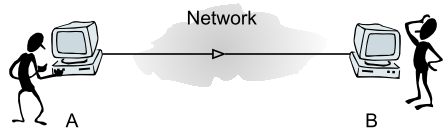


Figure 1.2: Information transfer from A to B

In the past organizations operating within a small private community used private networks to transport their information. These communities trusted each other and solved their problem internally, based on the long relationships built over the years.

There is a shift in focus however from private networks to more open network communication. Scaling-up, outsourcing, internationalization et cetera demand a collective infrastructure [dBC98]. The Internet appears at the horizon as a cheap media that can make easy and fast communication possible across the 'world'.

Connecting to the Internet has the problem that the quality of the processes, information systems and networks cannot be directly assessed from within the organization. This gives rise to the question whether information retrieved from the Internet is authentic and integer.

1.2.2 Increase usage of networks for doing business

In 1992 the number of hosts connected to the Internet reached one-million. By now, this number has increased dramatically. With so many users it is impossible to know everyone in person. The Internet, however, is being used to transfer a tremendous amount of information. The content of the information transferred across the Internet can be very different. While some information may be publicly distributed without any trouble, other information can contain classified information.

> *"Before using the Internet and integrating it into the business processes, the risks involved will have to be adequately assessed and controlled. There is a broad spectrum of quality aspects that has to be guaranteed will the Internet be useful for business specific processes"* [RvdB98].

Besides the quality aspects of information transfer, another aspect is also of great importance. This concerns the legal validity or legal usage of information exchanged across a network. While, for example, a paper based letter has legal value, none can be said of an e-mail message. After all, a digital document can be duplicated, changed or deleted without anyone finding out, and because the Internet consists of a huge amount of different networks, securing each network from tampering is an impossible task.

One of the aspects lacking an e-mail message is authentication. "We use authentication throughout our everyday life, for instance when we sign our name to some document. As we move to a world where our decisions and agreements are communicated electronically, we need to replicate these procedures" [RSA96].

1.3 Definition of the problem

This thesis will deal with the issues as portrayed above. The central theme of this thesis concerns the question:

> *What measures have to be taken to insure an open digital network*[3] *can be used to transfer information that complies with the feature of information transfer, as stated in section 1.2, and can be used as evidence in case of a legal dispute.*

In this thesis a distinction will be made between three areas of research. These concern the technical, organizational and judicial aspects. Because of the overall approach to these questions, the scope of this thesis has to be narrowed. Thus this thesis will only deal with those issues that are necessary to understand the working of the public key infrastructure. This will include the technical concept, that makes such an infrastructure possible, the infrastructure itself and the legislation, that sets a framework under which conditions the information sent can be used as evidence.

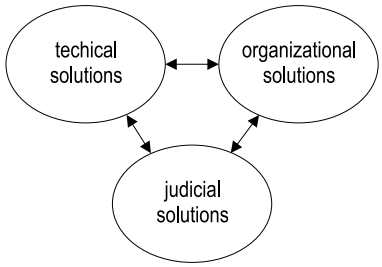


Figure 1.3: Different types of aspects for a public key infrastructure

1.4 Intent of thesis

This thesis has the intent of,

1. *providing requirements for secure information transfer across the Internet, that has such properties that the information sent has legally binding properties and*;
2. *testing current implementations against these requirements.*

This thesis is aimed at readers with a background in computer science or with affinity in computer science and issues involving the Internet and security. Most issues in this thesis are however readable for a broad public.

[3] With the term "open digital network" is implied an "interconnected collection of autonomous computers" [Tan96]. This thesis will further speak of the "Internet" instead of an open digital network, but what applies for the Internet also applies for an open network.

This thesis will not give a complete description of all the facets of the public key infrastructure. Only those issues that are important for a good notion of the working and the security of the public key infrastructure will be dealt with in this thesis. Many issues, such as the use of this media within an organization, will not be mentioned or not fully explained. The reader may however expect to know what constitutes a good digital signature that has legally binding properties and what the public key infrastructure constitutes.

1.5 Methodology

As part of my internship at *Deloitte & Touche*, I conducted research into different standards and legislation concerning the public key infrastructure.

Usage of standards under construction by the *PKIX working group* of the *Internet Engineering Task Force (IETF)* [AF98] and the *IEEE - P1363* working group [IEE98], has been used to develop insight into the current and ongoing development of the public key encryption infrastructure[4].

Extensive usage of the German law, [FMoET97] and [Sig97], has been used because this is a very rigid law that extensively discusses the technical requirements in several attachments, [Age97] and [Sch98]. Different legislation has further been used [Uta96], [N.V98] and the help of several books has improved my insight of these different legislation [Dut98], [Koo97] and [Koo98].

To develop insight into the working of several techniques extensive usage has been made of the RSA-laboratory that provides many information [RSA96].

Different articles have been used to provide extra information into those issues that where not clear to me.

The different requirements are then tested against quality aspects that have been formulated.

1.6 Contents

The content of the chapters is as follows.

Chapter 2 will first explain what risks-aspects are involved during an information transfer, with the help of a risk-model. The chapter will then explain how cryptography can insure network security. The focus of this chapter will be on public key encryption and the working of digital signatures.

Chapter 3 explains the organizational aspects concerning management of cryptographic keys. This will give an overview of all the actors involved and the functionality that is required for such an infrastructure. This will constitute the organizational aspects that have to be implemented.

Chapter 4 explains different legislation concerning the usage of digital signatures. This will constitute the judicial requirements that are necessary to obtain a digital signature that has legally binding properties. The usage of digital signatures is still

[4]The subscription to the mailing list of these groups has given me extra insight into current questions and new developments.

in its infancy. Legislators will still have to reach consensus on the issues involved. Even more difficult is reaching international consensus. An overview will be given of different trends in legislation and different legislation that has been implemented.

Chapter 5 will focus on specific aspects that have been mentioned, but not fully explained in the first chapters. Many of the issues in the chapter have a limited life-cycle, but they give an overview of current security threats and future threats that loom at the horizon. These will be explained in more detail.

Chapter 6 will give an case study of three current implementations. This chapter will explain how the different aspects have been implemented by these applications.

Chapter 7 the conclusion is a logical extension of the previous chapters This chapter will give a conclusion on the current status of digital signatures, the implementations, the security aspects and give recommendations.

Chapter 2

Network security using cryptography

2.1 Quality aspects of secure information transfer

> *"For the first few decades of their existence, computer networks were primarily used by university researchers for sending e-mail, and by corporate employees for sharing printers. Under these conditions, security did not get a lot of attention. But now, as millions of ordinary citizens are using networks for banking, shopping, and filling their tax returns, network security is looming on the horizon as a potentially massive problem"* [Tan96].

In the introduction, section 1.2 of chapter 1, several features of information transfer were introduced. These features constitute the quality aspects of information sent across a digital network. These features form the core of this thesis because if they can be assured then their is a basis for secure information transfer.

A business process depending on information, received from an information system, must be able to rely on this information with a great amount of assurance. If this information is received with the help of a network then a few aspects have to be assured.

These aspects form the *quality aspects of secure information transfer* and must be adhered to will it be possible to securely transport information across the Internet.

The definitions of these aspects are:

- *Authentication of sender and receiver,* the origin and the receiver of data should be irrefutably determined;
- *Non-repudiation of delivery and receipt,* the sender or receiver of data should not be able to deny having send or received a message;
- *Content integrity,* the content of the data send may not be susceptible to change or at least any change should be identifiable;
- *Confidentiality,* the content of data send should be illegible to third parties;
- *Availability,* the service should be available to authorized users;

- *Safe Usage,* the service should only be available to those users that are authorized to use it.

These items can be seen in figure 2.1. In this figure a company process depends on an information system. This information system is connected to another information system with the help of the Internet.

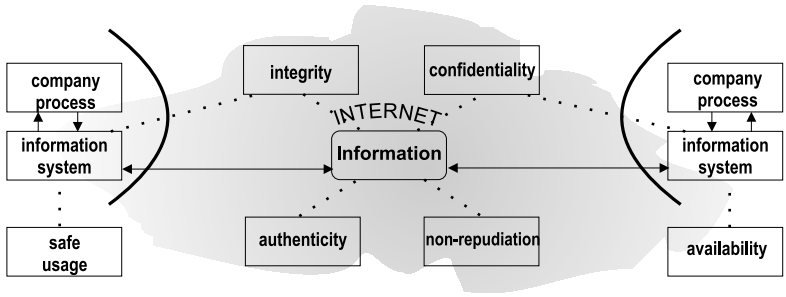


Figure 2.1: Quality aspects of information transfer [RvdB98].

The item availability will fall outside the scope of this thesis. Making the Internet available to a user can prove to be a rather complex. This thesis will however assume a user has access to the Internet.

If these aspects can be guaranteed with great amount of confidence then an open system network such as the Internet could be used to securely conduct business.

To get a better insight into the possible security measures at the different levels between the application system and the network, the levels between an application and the network as shown in figure 2.1 will be explained.

2.2 The OSI Reference Model

The hybrid TCP/IP - OSI reference model, as shown in figure 2.2, is a refined model based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various network layers [DZ83]. The model is called the *ISO OSI (Open System Interconnection) Reference Model* because it deals with connection of open systems – that is, systems that are open for communication with other systems, because they use the same framework.

The OSI model has seven layers. The principles that were applied to arrive at the seven layers are as follows:

1. A layer should be created where a different level of abstraction is needed;

2. Each layer should perform a well-defined function;

3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols;
4. The layer boundaries should be chosen to minimize the information flow across the interfaces;
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy.

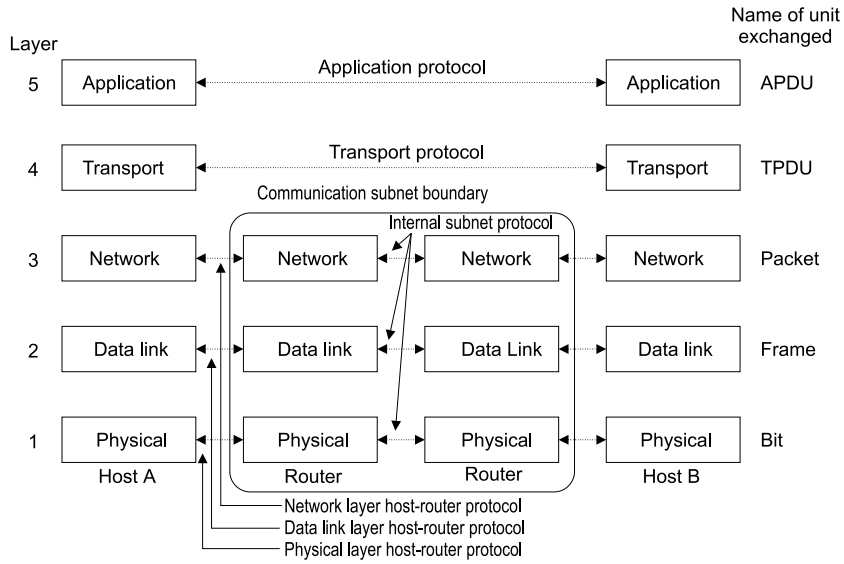


Figure 2.2: The hybrid TCP/IP – OSI reference model

At each layer of the OSI-model, a security feature can be implemented.

Security in the physical layer can be achieved by enclosing transmission lines in sealed tubes containing argon gas at high pressure. Any attempt to drill into a tube will release some gas, reducing the pressure and triggering an alarm. Some military systems use this technique [Tan96].

In the data link layer, packets on a point-to-point line can be encoded as they leave one machine and decoded as they enter another. All the details can be handled in the data link layer, with higher layers oblivious to what is going on. This solution brakes down when packets have to traverse multiple routers, however, because packets have to be decrypted at each router, leaving them vulnerable to attacks from within the router. Also, not all sessions have to be protected, e.g. sessions involving on-line purchases by credit card should be protected, while retrieving public information of a Web-site does not. Nevertheless, *link encryption*, as this method is called, can be added to any network easily and is often useful. In the network layer, firewalls can be installed to keep packets in or out.

Although these solutions help with secrecy issues and many people are working hard to improve them, none of them solve the authentication or non-repudiation problem in a sufficiently general way. To tackle this problems, the solutions must be in the application layer.

This chapter will first explain the security threats of a network and then the working of encryption including symmetric and public key encryption and the working of a digital signature.

2.3 Security threats of networks

At the start of this chapter, in section 2.1, a formal definition was given for quality aspects of secure information transfer. The problem concerns the situation as depicted in figure 2.3. Person A and B communicate with one another through a network. Because the network is so large it is virtually impossible to insure nobody can "tap into" the network.

Why can the above not be assured in an open network communication?

- someone standing in between a communication line can copy the signal that passes, thus the confidentiality is not met;
- someone can send a message saying: "I am A and this message is for B", thus the authentication is not met;
- someone can take a message, alter the contents and send it along, thus the integrity is not met;
- because of the above both A and B can claim not having sent a message and the non-repudiation is not met.

Why is the above possible? Because information sent across a network is sent digitally. This means it is not possible to unambiguously recognize an altered message as being altered[1]. Due to the large scale of networks it is furthermore virtually impossible to secure all network lines against tampering.

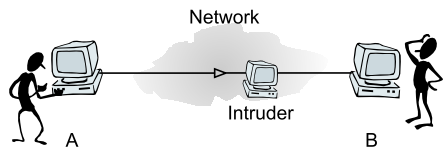


Figure 2.3: Intruder intercepting message

[1]This chapter will show that a message can be manipulated in such a way that a message can be unambiguously recognized and any change of the message is noticed.

2.3.1 How could a network be secured

Assuring the network against the problems, as stated in section 2.3, requires several measures to insure these aspects to be guaranteed. This section will take an intuitive approach to a solution. This solution will be given a more solid basis in the following sections.

This thesis will assume users communicate with help of the Internet. This implies these lines cannot be secured physically. Security should thus take place in a different manner. Because one should assume it is possible to copy a message sent across a network, the message itself should be made illegible for unauthorized people. This would require the message to be encoded in some sort of language only understood by the sender and the receiver.

Because the identity of a sender of a message can be "faked", the message sent should somehow be digitally signed. Due to ease of copying a message, it can be assumed there should be a relation between the message and the signature making it impossible to copy a signature and attach it to another message.

The following sections will explain how these requirements can be assured.

2.4 The working of encryption

Encryption is the transformation of data into some unreadable form. The purpose of encryption is to ensure privacy by keeping the information hidden from anyone for whom it is not intended, even those who can see the encrypted data. *Decryption* is the reverse of encryption; it is the transformation of encrypted data back into some intelligible form. Encryption and decryption require the use of some secret information, usually referred to as a *key*. Depending on the encryption mechanism used, the same key might be used for both encryption and decryption, while for other mechanisms, the keys used for encryption and decryption might be different [RSA96].

2.5 Encryption model

To get a better understanding of encryption and the actors involved, first the encryption model will be introduced, with the help of figure 2.4 [Tan96].

The messages to be encrypted, known as the *plaintext*, are transformed by a function that is parameterized by a *key*. The output of the encryption process, known as the *ciphertext*, is then transmitted. We assume that the *intruder* hears and accurately copies down the complete ciphertext. Because the intruder does not know what the decryption key is, he cannot decrypt the ciphertext easily.

Sometimes the intruder can not only listen to the communication channel, a so called *passive intruder,* but he can also record messages and play them back later, inject his own messages, or modify legitimate messages before they get to the receiver, a so called *active intruder*. The art of breaking ciphers is called *cryptanalysis.*

The art of devising ciphers, *cryptography* and breaking them, cryptanalysis, is collectively known as *cryptology*.

A fundamental rule of cryptography is that one must assume that the cryptanalyst knows how the encryption method works. The amount of effort necessary to invent, test and install a new method every time the old method is compromised or thought to be compromised has always made it impractical to keep this secret, and thinking it is secret when it is not does more harm than good.

This is where the *key* enters. The key consists of a (relatively) short string that selects one of many potential encryptions. In contrast to the general method, which may only be changed every few years, the key can be changed as often as required. Thus our basic model is a stable and publicly known general method parameterized by a secret and easily changed key.

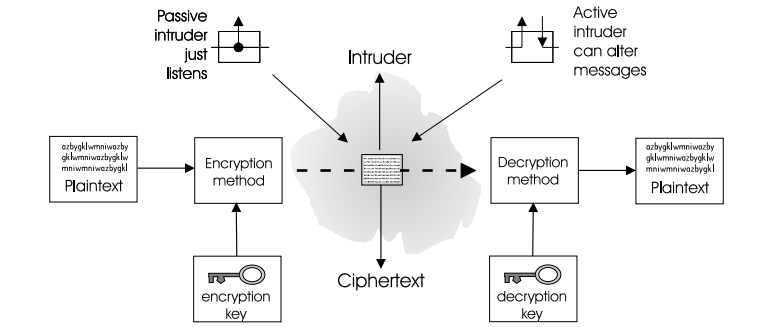


Figure 2.4: The encryption model [Tan96]

The real secrecy is in the key, and its length is a major design issue. Consider a simple combination lock. The general principle is that you enter digits in sequence. Everyone knows this, but the key is secret. A key length of two digits means that there are 100 possibilities. A key length of three digits means 1.000 possibilities, and a key length of six digits means a million. The longer the key, the higher the *work factor* the cryptanalyst has to deal with. The work factor for breaking the system by exhaustive search of the key space is exponential in the key length.

From the cryptanalyst's point of view, the problem has three principal variations.

- *ciphertext only*, here the cryptoanalist only has the ciphertext and no plaintext;
- *known plaintext*, here the cryptoanalist has both the plaintext and the ciphertext;
- *chosen plaintext*, here the cryptoanalist has the ability to encrypt pieces of plaintext of his own choosing.

Basically two distinct encryption methods exist:

- Symmetric encryption and
- Asymmetric encryption.

2.6 Symmetric encryption

Symmetric encryption uses an encryption method which uses the same key to encrypt information as well as to decrypt information.

Historically all cryptosystems were based on a system were the decryption key has a simple mathematical relationship with the encryption key.

If we use $C = E_K(P)$ to denote that the encryption of the plaintext P using key K gives the ciphertext C and similarly, $P = D_K(C)$ represents the decryption of C to get the plaintext again, then it follows that:

$$D_K(E_K(P)) = P \tag{2.1}$$

With symmetric encryption the key, D_K, to decrypt the ciphertext, C, is the inverse of the encryption key E_K.

$$D_K(E_K(P)) = P \quad \text{were} \quad D_K = E_K^{-1} \tag{2.2}$$

For a detailed discussion on symmetric encryption the reader is advised to read appendix A.

A widely known encryption algorithm that uses a product cipher[2] is *DES*. DES is the Data Encryption Standard, an encryption block cipher[3] defined and endorsed by the U.S. government in 1977 as an official standard. DES has been extensively studied since its publication and is the most well-known and widely used cryptosystem in the world [RSA96].

The requirements for symmetric encryption are:

- it should be impossible to calculate P from $E_K(P)$;
- it should be impossible to deduce E_K by any means;
- it should be possible to distribute E_K safely between the sender and the receiver of the message involved.

2.6.1 Drawbacks of symmetric encryption

Because all keys in a secret-key cryptosystem must remain secret, secret-key cryptography often has difficulty providing secure key management, especially in open systems with a large number of users [RSA96].

A further drawback of symmetric encryption is the one-on-one relationship between the sender and receiver. If a sender communicates with two people it will need two shared keys. If a sender communicates with three people it will need three shared keys. The total amount of keys in the infrastructure will however be, with n number of users:

[2] See Appendix A figure A.3 for a product cipher.

[3] "A block cipher transforms a fixed-length block of plaintext data into a block of ciphertext data of the same length. This transformation takes place under the action of a user-provided secret key. Decryption is performed by applying the reverse transformation to the ciphertext block using the same secret key. The fixed length is called the block size, and for many block ciphers, the block size is 64 bits" [RSA96].

$$1 + 2 + \ldots + (n-1) = \sum_{i=1}^{n-1} i = \frac{1}{2}(n(n-1)) \tag{2.3}$$

All these keys have to be stored and transported in a safe and secure way.

2.7 Public key cryptography

The key distribution problem has historically always been the weakest link in most cryptosystems. No matter how strong a cryptosystem was, if an intruder could steal the key, the system was worthless.

In 1976, two researchers at Stanford University Whitfield Diffie and Martin Hellman proposed a radically new kind of cryptosystem, one in which the encryption and decryption keys were different, and the decryption key could not be derived from the encryption key and vica versa [DH76].

In their concept, each person gets a pair of keys. These keys will be given the following names:

- *private key*, this key will be kept secret and will only be known to the owner;
- *public key*, this key will be kept in a public file.

These procedures have the following four properties:

1. Deciphering the enciphered form of a plaintext P yields P. Formally,

$$D(E(P)) = P \tag{2.4}$$

2. Both E and D are easy to compute.
3. By publicly revealing E the user does not reveal an easy way to compute D. This means that in practice only he can decrypt messages encrypted with E, or compute D efficiently.
4. If a plaintext P is first deciphered and then enciphered, P is the result. Formally,

$$E(D(P)) = P \tag{2.5}$$

The need for the sender and receiver to share secret information is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. No longer is it necessary to trust some communications channel to be secure against eavesdropping or betrayal.

2.7.1 One-way and Trap-door-one-way functions

An important factor for the working of public key cryptosystems are *one-way functions* and *trap-door-one-way functions.*

> "A *one-way function* is a mathematical function that is significantly easier to perform in one direction, the forward direction, than in the opposite direction, the inverse direction. It might be possible, for example, to compute the function in seconds but to compute its inverse could take months or years."
>
> "A *trap-door-one-way function* is a one-way function where the inverse direction is easy, given a certain piece of information, the trap door, but difficult otherwise" [RSA96].

The *one-way* function is a mathematical function that calculates from arbitrarily large data a unique fingerprint the outcome of which is called the *message digest*. More specific:

> Given an one-way function, H, that produces with plaintext p_1 a fingerprint f it is virtually impossible to find a plaintext p_2 that produces the same fingerprint f, or,

$$H(p_1) = f \wedge \nexists p_2 \rightarrow H(p_2) = f \tag{2.6}$$

The requirements for a one-way-function are:

- the plaintext, P, should be able to have a arbitrarily length;
- the output, the message digest, should have a fixed length;
- $H(p)$ should be easily to compute for an arbitrarily p;
- $H(p)$ should be one-way, e.g. it should be virtually impossible to compute p from $H(p)$;
- $H(p)$ is *collision-free*, e.g. knowing p_1 it should be virtually impossible to find a message p_2 such that $H(p_1) = H(p_2)$.

As an example for a *trap-door-one-way function* one can think of exponentiation. Exponentiation can be computed easily by repeatedly multiplying. The inverse of exponentiation, calculating the square root, is far more difficult to compute. One of the most well known and used public key systems is the *RSA-system.*

For a more detailed discussion on asymmetric encryption the reader is advised to read section 5.3 of chapter 5 or appendix B.

As stated in the introduction, encryption is more then encrypting and decrypting. Another fundamental part of our lives is authentication. The next section will discuss how a digital signature can achieve this.

2.8 Digital signature with the use of the PKI-system

Devising a digital signature requires a combination of information concerning the message itself and that of the signer. Because digital information can be cut copied and pasted, there should be a link between the message and the digital signature itself. Otherwise, the recipient could modify the message before showing the message-signature pair to a judge. Even worse, he could attach the signature to any message whatsoever, since it is impossible to detect electronic 'cutting' and 'pasting'.

The following desirable properties can therefore be deduced from a digital signature:

- The signature is message dependent;
- Only the originator of an electronic message can compute the correct digital signature;
- Anyone who receives a message and a digital signature can verify the signature and consequently be certain of the origin and integrity of the message.

This is were the one-way function enters. The one-way function can make a unique fingerprint of a message. This unique fingerprint is then encrypted with the trap-door-one-way function. This is called the *digital signature*. The signature is message dependent because of the unique digital fingerprint and it is uniquely bound to the issuer because of the encryption with the unique private key.

For our scenarios we suppose that A and B (also known as Alice and Bob) are two users of a public-key cryptosystem. If we denote E_x as the encryption key of X, then we will distinguish the encryption and decryption procedures of A and B with: E_A, D_A, E_B, D_B.

If Alice wants to send a signed message to Bob, then the digital signature, as shown in fig 2.5, is computed as follows:

- Of the plaintext a unique fingerprint is calculated with the help of a one-way function, $H(p)$.
- The result of the calculation is then "signed" with her private key D_A. Thus the digital signature now consist of: $D_A(h(p))$.
- Alice then sends the plaintext, P along with her signature, $D_A(h(p))$, to Bob.

Bob will do the following after receiving the message:

- Of the plaintext he will calculate the unique fingerprint with the help of the same one-way function, $h(p)$.
- He will then decrypt the unique fingerprint with the help of the public key of Alice, E_A, which is available in the public file, $E_A(D_A(h(p)))$.
- Bob will then compare the value of his calculation with the value calculated by Alice. If the two match then the message has been signed by Alice and the message is unaltered.

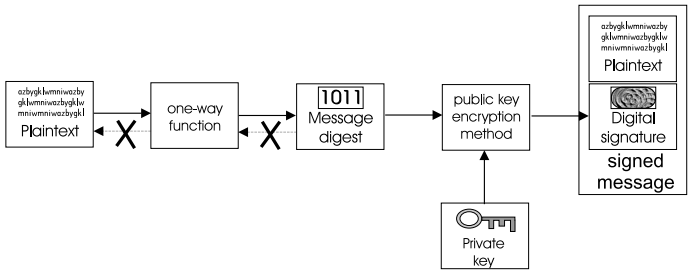


Figure 2.5: Digital signature attached to plaintext

Alice cannot later deny having send message, P, because only she could have signed the message. Furthermore, she, or anyone else, can not modify plaintext P because a message P' would produce a different signature.

Bob on the other hand cannot use the signature for any other message because it is unique.

2.9 Public key encryption can provide secure solutions

To prove that public key encryption can provide a solution for secure information transfer the following subsections will show how the quality aspects, as stated in section 2.1, can be complied with. This will only include those issues that involve the transfer of information across an open network. Issues involving the safe usage will not be dealt with at this point.

2.9.1 Authentication and safe usage

If it can be assured, that the private key of a person only stays with one person and this person is the only having access to this key, then the following occurs: because that person is the only person in the possession of the private key, only he could have made such a message. Because the private key is uniquely linked with the corresponding public key, the message is unique if they link by decrypting it with the accompanying public key. The message must then also have been written by this person. Thus the authentication is met.

2.9.2 Non-repudiation of delivery

Because the document has properties that make it authentic, the non-repudiation of delivery is met. If someone can produce a document which is authentic then the non-repudiation of delivery is proofed.

2.9.3 Non-repudiation of receipt

Non-repudiation of receipt is a difficult aspect. When has a message been received? It can only be proven if a confirmation is sent which states that the message has been received. Thus a confirmation should be sent that has been encrypted with the private key of the receiving party, or or in other words a signed confirmation should be sent.

2.9.4 Content integrity

Because the message is encrypted it cannot be read by other people but the contents cannot be changed because otherwise the content will become meaningless. Thus if the message is readable then the contents are also integer.

2.9.5 Confidentiality

If Alice wants to send a confidential message to Bob then she will do the following:

- Retrieve the public key of Bob from the public file, E_B.
- Encrypt the plaintext, P: $E_B(P)$.
- Send the encrypted message to Bob.

When Bob receives the message he will decipher the message by computing $D_B(E_B(P))$. By property 3 of the public-key cryptosystem[4] only he can decipher the message because he is the only person in the possession of the private key. He can also encipher a private response with E_A, that is also available in the public file.

The beauty of the system is that no prior contact is necessary to establish private communication[5].

2.9.6 Drawback of Public Key cryptography

An "RSA operation," whether for encrypting or decrypting, signing or verifying, is essentially a modular exponentiation, which can be performed by a series of modular multiplications. These multiplications require a considerable amount of time compared to symmetric encryption techniques such as DES. By comparison, DES is much faster than RSA. In software, DES is generally at least 100 times as fast as RSA. In hardware, DES is between 1,000 and 10,000 times as fast [RSA96].

Nevertheless, public-key cryptography can be combined with secret-key cryptography to get the best of both worlds.

2.9.7 Digital envelop

To increase the speed of encryption, the best solution is to combine public and secret key systems in order to get both the security advantages of public-key systems and the speed advantages of secret-key systems.

This can be achieved in the following way:

[4] By publicly revealing E the user does not reveal an easy way to compute D, see section 2.7

[5] A few questions still remain unanswered. For instance one of the problems remaining that will be tackled, in chapter 3, concerns the validity and authenticity of the public key.

1. A symmetric key is generated;
2. The symmetric key is then used to encrypt the bulk of a file or message;
3. The symmetric key is then encrypted with the help of the public key system;
4. A digital signature is then made of both the ciphertext and the symmetric key;
5. The ciphertext, the encrypted symmetric key and the digital signature are then put together in what is known as a *digital envelop.*

This can be seen in figure 2.6.

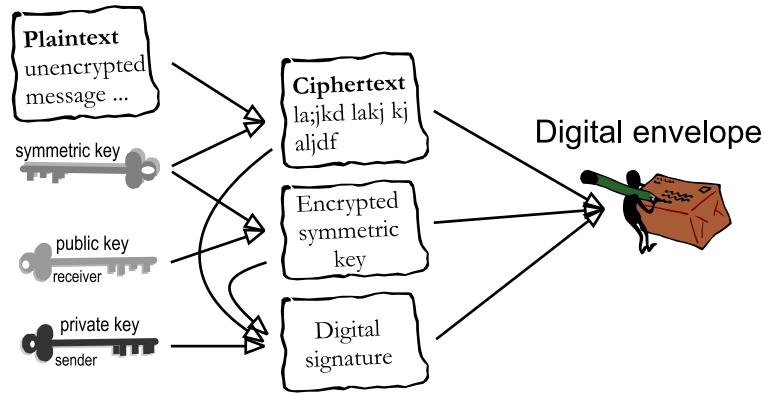


Figure 2.6: Digital envelope

The advantages are speed. Symmetric key encryption is much faster that public key encryption. Because both systems are equally “strong” in terms of security the systems is both secure and fast.

2.10 Summary

In this chapter it was first stated what security should involve. This was built up of the requirements that should be met. Secondly a short introduction into symmetric key encryption was given and an introduction into public key encryption. Third it was shown how a digital signature is made. finally it was shown that the requirements as stated at the start of the chapter were fulfilled apart from the safe-usage.

Technically there thus exists a way to secure information sent across the Internet. Issues that will however have to be resolved include safe usage of keys.

Chapter 3

Public Key Infrastructure

As shown in the previous chapter, the requirements for secure information transfer can be achieved with the help of public key encryption.

These requirements, that have already been stated in the previous chapter, are stated below:

1. *Authentication of sender and receiver*
2. *Non-repudiation of delivery and receipt*
3. *Content integrity*
4. *Confidentiality*

The basis for the secure information transfer consists of the private/public key-pair. The private key only known by the owner and the public key made available to other users who need to have this key.

At this point users are exchanging their public keys somehow. They then securely exchange a message by encrypting the messages with help of the public key, they somehow obtained. This leaves a few questions unanswered. To guarantee a proper working of the public private key system, several requirements have to be assured. Requirements that have to be assured are:

- A person/entity should be able to securely retrieve the public key of another person/entity and be assured that the public key he obtained *is* the public key of that person;
- A person/entity should be able to securely assess the validity of the public key of another person/entity.

This chapter will explain how these requirements can be met and what procedures are required.

3.1 Management of public keys

To avoid tampering with public keys the public keys are placed in a file that is digitally signed by, what is known as a *trusted party*[1]. The name and public key of a person/entity together with the digital signature of the trusted party is called a *Certificate*.

> *"A certificate is a digital document attesting to the binding of a public key to an individual or other entity"* [RSA96].

Once signed by a trusted person, a certificate is secured against tampering and can be distributed by disk, network or even written on a piece of paper. A common procedure is to have a certificate server connected to a network. This can be seen in figure 3.1. The certificates are stored on a server connected to a network from which they can be obtained.

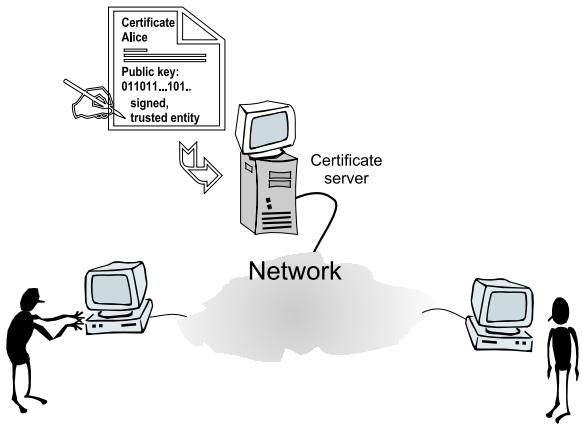


Figure 3.1: Certificate server connected to network

If users now wish to communicate privately then they can do so by first exchanging their certificates. Because the certificates are digitally signed, by a trusted party that *both* users trust, they can be freely distributed without being tampered with unnoticed. Any change will be made visible with the help of the digital signature that has been attached.

Verifying the certificate can be done by anyone who is in the possession of the certificate of the trusted party. With the certificate of the trusted party, a person can check the certificate and thus check the digital signature for correctness.

[1] A trusted party is a person or object that has certain properties that make it '*trusted*'. This issue will become more clear in the following sections.

3.1.1 Contents certificate

Because the certificate is so important for authentication, defining meaningful contents of a certificate is essential. As a minimal requirement of the contents of the certificate, one can think of the items listed in table 3.1.

Name of the field	*Contents and functions*
Name	Distinguished name of the authenticated subject or entity
Key	Subject or entity's public key information: Algorithm, parameter, key.
Signature	Overall signature by the certification authority. The signature of the certification authority binds the public key to the entity's name

Table 3.1: Minimum contents of a certificate

An essential, but difficult task is to assign meaningful content to the *name*, or to give a *distinguished name*. In practice it proves hard to give meaningful content. A name should give a unique binding to an entity[2].

A well known system that uses the public key encryption technique and certificates is *Pretty Good Privacy* or *PGP*.

3.2 Pretty Good Privacy

PGP is a milestone in the history of cryptography, because for the first time it made cryptography accessible to the wide mass of privacy hungry on-line public.

PGP was created primarily for encrypting e-mail messages using public or conventional key cryptography. The latter are used mainly to encrypt local files. With public key cryptography, PGP first generates a random symmetric session key and encrypts the plaintext with this key. The session key along with the ciphertext are then encrypted using the recipients public key and then forwarded to the recipient[3].

Other features include generating message digests, generating digital signatures, management of personal "key rings[4]" and distributable public key certificates. It is also designed to work off-line to facilitate e-mail and file encryption, rather than on-line transactions [AR96].

3.2.1 Certification with PGP

PGP's uses the "*web of trust*" approach to generate secure certificates. In this approach, there are no central authorities which everybody trusts, but instead, individuals sign each others key and progressively form a web of individual public keys interconnected by links formed by these signatures.

[2]This will be further clarified in chapter 5

[3]This is also called a digital envelop as explained in the previous chapter.

[4]A key ring is a link of certificate of other persons. Before communicating with other people the certificates of those other people have to be obtained and stored for later use. This is called a "key ring".

In this method if Alice trusts Bob and Bob trusts Carol then Alice will trust Carol because she trusts Bob. More formally, if we use the notation $A_t(B)$ to denote Alice trusts Bob then the following occurs:

$$A_t(B) \wedge B_t(C) \Rightarrow A_t(B_t(C)) \Rightarrow A_t(C) \tag{3.1}$$

3.2.2 Trustworthiness of public-key certificate

The certificates within the PGP infrastructure are further refined by adding extra information concerning the trustworthiness of the certificate and the trustworthiness of the certificate to introduce another certificate.

There are roughly three categories of confidence in a certificate defined in PGP. These are as follows [AR96]:

1. *undefined*, we cannot say whether this public key is valid or not;
2. *marginal*, this public key *may* be valid but we cannot be too sure;
3. *complete*, we can be wholly confident that this public key is valid.

The confidence in certificates is further extended by adding information concerning the trust in a certificate to introduce another certificate. More formally if Alice trusts Bob this does not automatically mean Alice will trust Carol because Bob does or:

$$A_t(B) \wedge B_t(C) \not\Rightarrow A_t(C) \tag{3.2}$$

These trust levels have been divided into four levels:

1. *full*, this public-key is fully trusted to introduce another public-key;
2. *marginal*, this public-key can be trusted to introduce another public-key, but, it is uncertain whether it is fully competent to do that;
3. *untrustworthy*, this public-key should not be trusted to introduce another, therefore any occurrence of this key as a signature on another public-key should be ignored;
4. *don't know*, there are no expressions of trust made about this public-key.

Introducing concepts such as trustworthiness and trust as to whether a certificate can introduce another certificate, cannot overcome the problems associated with finding out whether a certificate is authentic and the contents is correct. This lack of fixed or formal certification paths means that the uncertain authenticity of any PGP key certificate becomes a rather significant matter [AR96].

Thus the use of PGP is limited to a small community of users that 'know' each other. For business practices this does provide a good solution. Furthermore it does not provide for a solid bases in case of a legal dispute.

3.3 Trust infrastructure

Trust can be defined as:

> *"the confidence in a person or thing because of the qualities one perceives or seems to perceive in him or it"* [Lex].

Currently PGP communication partners are able to exchange keys that allow for trusted communication within a small community. But they are not in the position to authenticate unknown partners, especially in an open infrastructure with communication partners previously unknown.

In an open and large-scale network, it is impractical and unrealistic to expect each user to have previously established personal or physical relationships with all the expected communication partners [Poh97].

To establish a trusted communication path in a modern communication infrastructure, a sender of an object, a message or a data file, must be able to identify and authenticate the receiver, the business or communication partner, reliably without having to meet him personally in order to trust him.

3.3.1 Trusted Third Party

The generally accepted way today to authenticate a new communication partner or a new receiving entity is to authenticate him by a third authority or party. This concept allows that two individuals implicitly trust each other although they have not previously established a personal relationship. This is shown in figure 3.2. In this scenario, the guarantee for the correct identity is provided by a third party that assures to each of the communication partners that the other partner is authentic, or more formally:

$$A_t(T) \wedge T_t(B) \Rightarrow A_t(B) \tag{3.3}$$

Such a party that has to be trusted by all the other entities participating in the information exchange is called a *Trusted Third Party (TTP)*. Because a TTP issues certificates it is often referred to as a *Certification Authority* or *CA*.

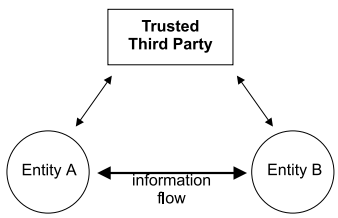


Figure 3.2: Generating trust with a Trusted Third Party

3.3.2 Actors within a TTP-model

For a clear understanding of the TTP model, all the actors involved will first be explained [AF98]:

- *End-Entity.* This term is used to refer to the entity named in the subject field of a certificate. It is important to note that the end entities here will include not only human users of applications, but also applications themselves (e.g., for IP security);
- *CA* or *Certification Authority.* We use the term CA to refer to the entity named in the issuer field of a certificate. The certification authority (CA) may or may not actually be a real "third party" from the end entity's point of view. Quite often, the CA will actually belong to the same organization as the end entities it supports;
- *RA* or *Registration Authority.* In addition to end-entities and CA's, many environments call for the existence of a Registration Authority (RA) separate from the Certification Authority. The functions that the Registration Authority may carry out will vary from case to case but may include personal authentication, token distribution, revocation reporting, name assignment, key generation, archival of key pairs, et cetera.

3.3.3 Services of a trust infrastructure

As stated before, one of the services of a Trusted Third Party will be to attest to the identity of a user. This attest will be given in the form of issuing a certificate that has been signed with the private key of the CA.

Other services that must be carried out by the TTP include storing, distributing, updating, revoking certificates and, one of the more important service, attesting to the validity of a certificate. These issues will be explained in the next sections.

3.3.3.1 Issuing certificates

A certificate authority issues certificates for an entity. By issuing a certificate, the CA attest to the certificate as being trustworthy. Depending on the practices, carried out by the CA, to insure the contents of the certificate, certain assumptions can be made about these contents.

To improve the trustworthiness of a certificate extra information can be added. In table 3.1.1 the minimum contents for a certificate were given.

According to German law the contents of a certificate should include [FMoET97]:

- name of the owner of the signature key to which additional information must be appended in the event of possible confusion;
- public signature key assigned;
- names of the algorithms with which the public key of the owner of the signature key and the public key of the certification authority can be used;

- serial number of the certificate;
- beginning and end of the validity period of the certificate;
- name of the Certification Authority and
- an indication as to whether use of the signature key is restricted in type or scope to specific applications.

This first item already poses some question. “The name of the owner to which additional information must be appended in the event of possible confusion”. A name can hardly be unique. Therefore extra information should almost certainly be given. This information could be given in the form of a social security number, credit card number, or maybe even a photo image in jpeg format, et cetera. This extra information could also be given in information concerning the employment. This could concern company name, function or even creditworthiness.

3.3.3.2 Distributing certificates

Because a certificate has been signed by the TTP, any tampering can be securely assessed. Distribution can be achieved by sending it by ordinary e-mail or storing it in a publicly available file.

3.3.3.3 Validating certificates

An important issue concerning certificates is the validity of each certificate. Trusting a certificate while the owner might have lost the private key or the contents have become obsolete should be prevented. A common procedure often used is the issuance of a *Certification Revocation List* or *CRL*.

A CRL is a list issued by a CA that contains all certificates that have been revoked. If an entity wishes to check the status of a certificate he has to retrieve the CRL and check whether it is on this list or not.

One of the problems concerning the CRL is the difference in time between a certificate being revoked and the new CRL issued. This means there can be a considerable time between a certificate being revoked and a person being able to verify this. In cases of the transfer of highly sensitive information this can be considered not secure enough. Another problem concerns the sheer size of the CRL list. This list can grow significantly as the amount of certificates being revoked increases.

This has given rise to another method of checking the status of the certificate, the *Online Certificate Status Protocol* or *OCSP*. The OCSP gives real-time information concerning the status of a certificate. The following status can be given: notRevoked, revoked, onHold and expired.

3.3.3.4 Revoking a certificate

A user whose private key has been compromised should revoke his certificate to stop any misuse. Procedures have to be set up under which conditions a certificate can be

revoked, who has the authority, what further measures have to be taken or what legal consequences this can have.

Thus the time-gap between the revocation request and the actual moment of revocation is very important. The longer the time-space the more damage can be done. A CRL and the time between CRL-updates thus becomes very important.

These functions as explained above can be seen in figure 3.3. This figure has been divided into *PKI-users* and *PKI-management entities*, thus making a distinction between management and end users. These items have been number and have the following functions [AF98]:

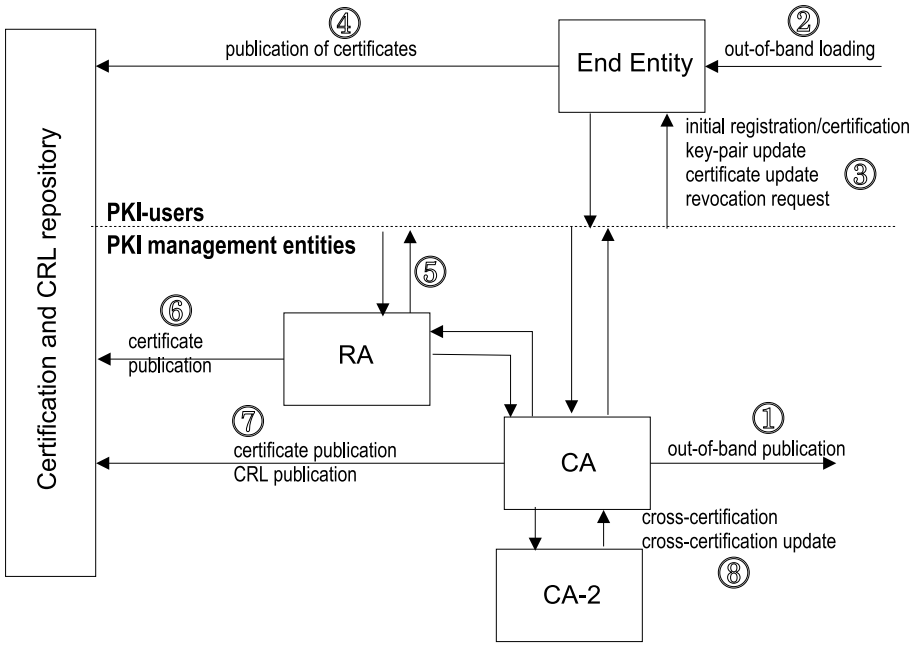


Figure 3.3: Trust infrastructure [AF98]

1. *Out-of-band publication*, the CA will have to make its certificate public. With this certificate end-entities can securely send a message to a CA, for instance a request to a certifcate[5];
2. *Out-of-band loading.* The end-entity will have to securely obtain the certificate of the CA. This will allow him to send a secure message to the CA to request a certificate;

[5]the out-of-band publication refers to the fact that there is not one special way to publish a CA-certificate. Publication can thus occur by newspaper, diskette network or the certificate can be incorporated into a web-browser as is very often the case with Netscape and Internet Explorer.

3. This step has multiple actions, these include:

   (a) *Initial registration/certification*. This is the process whereby an end entity first makes itself known to a CA or RA, prior to the CA issuing a certificate or certificates for that end entity. The end result of this process (when it is successful) is that a CA issues a certificate for an end entity's public key, and returns that certificate to the end entity and/or posts that certificate in a public repository. This process may, and typically will, involve multiple "steps", possibly including an initialization of the end entity's equipment. For example, the end entity's equipment must be securely initialized with the public key of a CA, to be used in validating certificate paths. Furthermore, an end entity typically needs to be initialized with its own key pair(s);

   (b) *Key pair update.* Every key pair needs to be updated regularly (i.e., replaced with a new key pair), and a new certificate needs to be issued;

   (c) *Certificate update.* As certificates expire they may be "refreshed" if nothing relevant in the environment has changed;

   (d) *CA key pair update.* As with end entities, CA key pairs need to be updated regularly; however, different mechanisms are required;

4. *Publication of the certificates.* Having gone to the trouble of producing a certificate, some means for publishing it is needed. This could include a certificate server that is attached to a network;

5. Certain functions of the CA can be conducted by the RA. These include those as stated in item 3;

6. *Publication of the certificates.* This is the same as stated previously;

7. *CRL-publication.* If a certificate somehow becomes compromised that this will have to made known. This will be done with the CRL-publication that is updated periodically;

8. *Cross-certification* and *Cross-certification-update*. If a users wishes to communicate with another person that has a certificate issued by a another CA, CA_2, then a cross certificate can be made that will allow an end-entity to verify the certificates issued by CA_2.

All the services and operations as stated above are part of the *key management.*

3.4 Types of TTP

A distinction can be made between three types of TTP's, as shown in figure 3.4:

1. *Off-line TTP*. An off-line TTP does not interact with the user entities during the process of the given security service. Instead the interaction to provide, or register, security-related information is carried out off-line as a separate interaction. The results of such an interaction may be cached and reused to avoid having to communicate with the server each time communication is initiated;

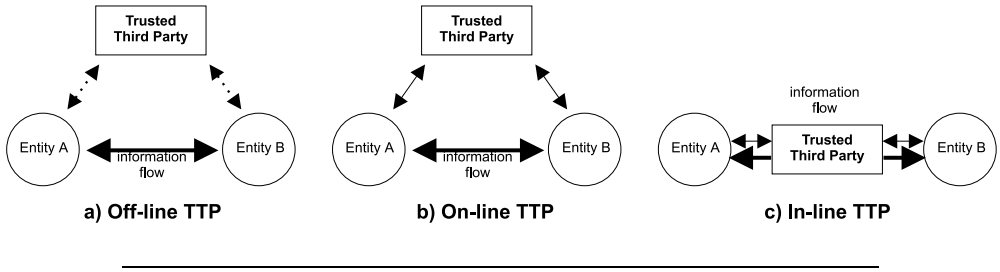


Figure 3.4: Different types of TTP's

2. *On-line TTP*. An on-line TTP is requested by one or both entities in real-time to provide, or register, security-related information. Such a TTP is not in the communications path between the two entities;

3. *In-line TTP*. An in-line TTP is positioned in the communication path between the entities. Such an arrangement allows the TTP to offer a wide range of security services directly to users. Since the TTP interrupts the communication path, different security domains can exist on either side of it.

3.5 Public Key Infrastructure (PKI) Models

There are two principle PKI models, the "Open PKI Model" and the "Closed PKI Model" [McC97].

3.5.1 Open PKI Model

The Open PKI Model is a public model in which a CA provides "generic" all-purpose digital certificates to the subscriber. The relationship with the individual requesting the digital certificate is typically minimal and the subscriber vouches for her own identity. The issued digital certificate does not provide the recipient with much information about the certificate holder.

3.5.2 Closed PKI Model

Under the Closed PKI model, digital certificates are issued for a specific purpose by an organization or business that has an established relationship with the subscriber (such as an employer issuing digital certificates to its employees). The business or organization determines the verification level[6] necessary for issuing the digital certifi-

[6]For example VeriSign, currently one of the major issuers of digital certificates, supports three distinct certificate classes. Each class provides for a designated level of trust. The differences concern the amount of verification before the certificate is issued. For example Class 1 certificates confirm that a users name (or alias) and e-mail address form an unambiguous subject name within the VeriSign repository, whereas Class 3 certificates provide important assurances of the identity of individual subscribers by requiring their personal (physical) appearance before a Class 3 LRA or its delegate (such as a notary) [Ver97].

cate, depending on its intended use. The specific use for which the digital certificate is issued could be things such as electronic payments or secure network access.

3.6 Certification practice statement

Different security requirements call for different security measures. The transfer of one million dollars is probably a more sensitive information transfer then a simple happy birthday message to a friend.

The most common procedure to check for the "trustworthiness" is to check for the policy statement of the TTP. This is usually referred to as the *Certification Practice Statement* or *CPS*.

> *"A Certification Practice Statement is a statement of the practices which a certification authority employs in issuing certificates"* [Ass96].

Reading a CPS can give information concerning the operations of a CA. This most often concerns the practices conducted by the CA. It can be seen as a contract between an end-entity and a CA.

Trusted Third Parties can be established independently for different applications, business sectors or geographical regions. However there is a need for cooperation if an information exchange between these different areas is required like, for example, for electronic commerce. The technical basis for this co-operation is called a *trust infrastructure* [Poh97].

This trust infrastructure can only be established if the same practices are implemented at both TTP's. If differences exist in the interpretation of names or functions then the trustworthiness is in jeopardy. Thus the CPS of two TTP's willing to cooperate should equal. Equal policies can make a partnership possible. A method to achieve this is with the help of a *Policy Authority*.

3.6.1 Policy Authority

To grant trust to the users of a trust infrastructure, it is necessary to establish trustworthy TTP's. Otherwise the users may not be sure that they know either the user they communicate with nor the TTP as the issuer of the certificate for the correspondent user [Poh97]. A method to achieve this is to establish a policy certification authority. A policy authority could certify a TTP and issue a certificate that authorizes the TTP.

The following types of authorities can be distinguished [Poh97]:

- *Policy Approval Authority (PAA)*
  An authority which establishes the overall infrastructure security policy and creates guidelines that all subordinate entities must follow. The PAA also acts as a root certification authority, issuing certificates for the next tier of certification authorities (PCA's).

- *Policy Certification Authority (PCA)*
  An authority which establishes policy for a single organization or single community of interest. A PCA also acts as a certification authority for the next tier of certification authorities (CA's).

- *Certification Authorities (CA)*
  An authority trusted by one or more users to create, assign and issue public verification key certificates to end entities and other certification authorities certificates - by binding the public key and an entity - may be an individual - by name. Optionally the certification authority may create the users keys. Certification authorities issue certificate revocation lists periodically, and post certificates and certificate revocation lists to a repository.

- *Registration Authority (RA)*
  An entity that acts as an intermediary between the CA and a prospective certificate subject; the CA trusts the RA to verify the subject's identity and that the subject possesses the private key corresponding to the public key to be bound to that identity in a certificate.

These actors can be seen in figure 3.5. In this figure there is a Policy approving authority that issues the general guidelines. Those complying with the overal guidelines form a Policy Certification authority that establish policy for single organizations or communities that have the same interest, such as hospitals, banks et cetera. Different CA's that comply with these policies can then be formed. The actual registration can then be performed by an internal registration authority within a company.

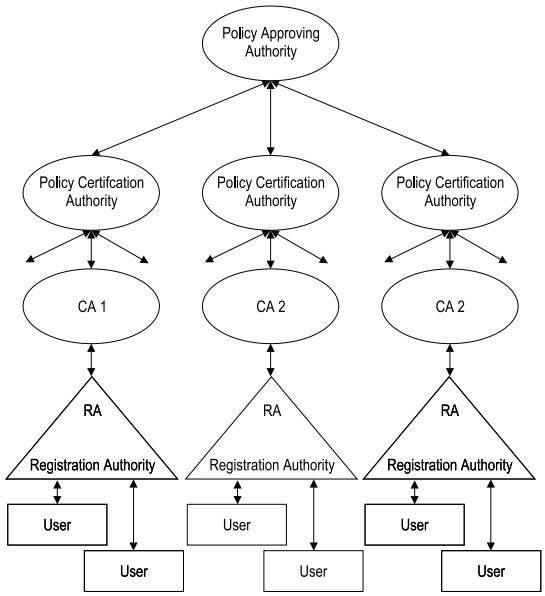


Figure 3.5: Hierarchical Trust Model [Poh97]

Once the policy approves then alliances can be formed. This can be done in several ways as shown in figure 3.6. This may include [Poh97]:

- *Single Centralized Authority.*

A centralized architecture consists of only one centralized authority. Such a centralized authority is relatively inflexible and does not scale well in an environment where different sectors, different application areas and different countries might be involved. It seems also difficult for political reasons to promote such a centralized authority if different member states are involved.

- *Hierarchical Certificate Model*
  Authorities are arranged hierarchically under a "root" certification authority that issues certificates to subordinate certification authorities. These certification authorities may in turn issue certificates to subordinate certification authorities, or to users. Every user knows the public key of the root certification authority, and any user's certificate may be verified by verifying the certification path that leads back to the root certification authority.

- *Network Certificate Model*
  Independent certification authorities cross-certify each other, resulting in a general network of trust relationships between certification authorities. A user knows the public key of a certification authority near himself, generally the local certification authority that issued his certificate, and verifies certificates by verifying a certification path that leads back to that trusted certification authority.

- *Hybrid Certificate Model*
  The hierarchical and network trust infrastructure architectures are not mutually exclusive. The following is a hybrid certification path architecture: There will be a hierarchical path of certificates leading from the root certification authority to its subordinate certification authorities, and from each of these certification authorities to their subordinates, and so on, until every end user is issued a certificate with a certification path from the root certification authority. Each certification authority will have a single parent. In parallel to the certificates hierarchically linking certification authorities to the root will be cross-certificate pairs attributes also linking those certification authorities. These parallel cross-certificate pairs are required. This will allow client applications that perform certification path verification from the verifier's parent certification authority, using the cross-certificate pair directory attribute, to operate from any certification authority. Certification authorities may cross-certify each other along paths that do not parallel the hierarchy.

3.7 Summary

The previous chapter showed how a technical solution could be implemented that made it possible to securely send information across the Internet.

The organizational aspects necessary to distribute information to end-entities was explained in this chapter. It was shown how a public key can safely be distributed, with a certificate. Furthermore the management of certificates with the help of a certification authority was explained. To manage certificates certain functions are necessary, thus constituting the functions of a CA. The operations of a CA are then written down in a

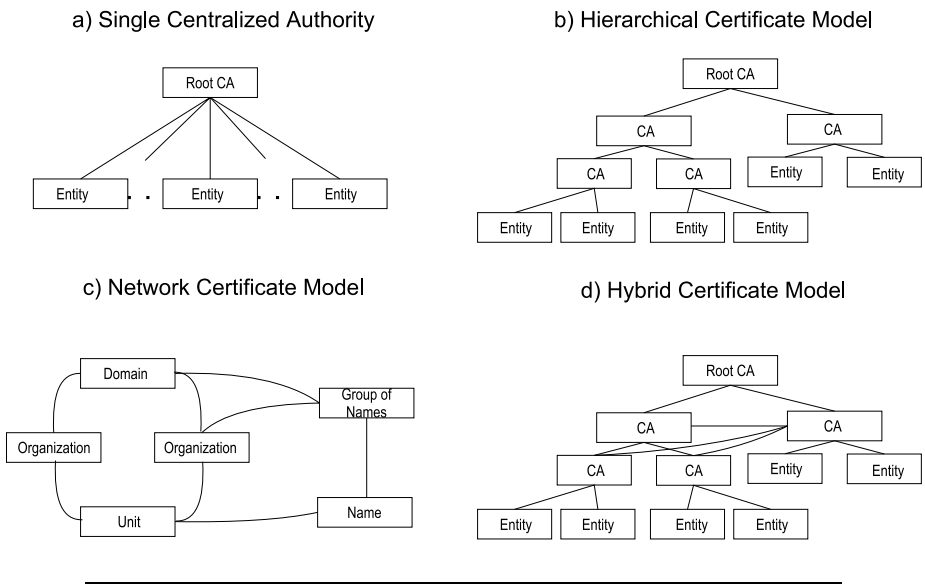


Figure 3.6: Options for a trust infrastructure

Certification Practice Statement, that can be seen as a contract between an end-entity and a CA. By matching the operations of different CA's, these can start to co-operate. How this can be done was explained at the end of this chapter.

At this moment it has been shown how technical and organization measures, within the boundaries of this thesis, can insure secure information transfer. Still remaining is the question whether the information sent in the way as described in the previous chapter possesses legal validity.

Chapter 4

Legislation for the public key infrastructure

> *"The formal requirements for legal transactions, including the need for signatures, vary in different legal systems, and also vary with the passage of time. There is also variance in the legal consequences of failure to cast the transaction in a required form. The statute of frauds of the common law tradition, for example, does not render a transaction invalid for lack of a writing signed by the party to be charged, but rather makes it* unenforceable *in court, a distinction which has caused the practical application of the statute to be greatly limited in case law"* [Ass96].

The above statement describes the problem of electronic commerce. Electronic commerce can be conducted across the Internet but if a dispute arises, then the evidence is unenforceable in court, because the legal requirements of information sent across the Internet are near to non-existent.

The (inter)national business world and international organizations have identified several legal obstructions that obscure electronic commerce [Act98]:

- Indistinction concerning the jurisdiction: the international field of electronic commerce leads to the question concerning what law has to be applied, for instance, in case of a legal dispute;
- Uncertainty concerning the liability of the intermediary: it has to be made clear what the status of the intermediary is in case of an unauthorized action with electronic commerce;
- Indistinction concerning the digital signature: an infrastructure to issue digital signatures, has not been developed as of yet due to uncertainty concerning the judicially status, including the legal validity;
- Uncertainty concerning the storage and prove of evidence: the necessity of a uniform storage-obligation, due to the increased intensity of electronic information transfer, has increased and become more important. A question concerns the requirements that apply for the prove of an electronic data transfer;

- Legislation in existing laws can hinder the usage of electronic data transfer.

4.1 Legislation for digital signatures

A great many countries have enacted or are considering to enact digital signature legislation in an effort to facilitate electronic commerce. Current legislation can be broken down into three models, which will be called the *Limited Legislative Model, the Comprehensive Legislative Model* and the *Minimalist Legislative Model* [McC97]:

1. *Limited Legislative Model*
   The term, "Limited Legislative Model" will refer to digital and electronic signature legislation that is very narrow in scope, covering only such things as government communications, electronic filing of securities-related documents, Certificates of Death and voter registration.

2. *Comprehensive Legislation*
   The term "Comprehensive Legislation" will be used to refer to an all-encompassing regulatory approach that pertains to all communications, and provides for much, although not necessarily all, of the following: regulates certification authorities, prescribes duties of CA's and subscribers, sets forth warranty, liability and limitation of liability provisions, is technology-specific (i.e., covers only digital signatures), sets forth the legal effects of digital signatures and electronic message and presumptions in adjudicating disputes, establishes a state agency as a CA and establishes repositories and prescribes their liability.

3. *Minimalist Legislative Model*
   The term "Minimalist Legislative Model" will be used to refer to legislation that pertains to all communications, is technology-neutral and basically does little more than give legal effect to electronic signatures and electronic records.

"Although the express purpose of the Limited Legislative Model may not necessarily be to encourage electronic commerce, nonetheless, it will likely achieve this objective because it provides for specific situations in which people can become accustomed to filing documents and communicating with governments electronically" [McC97].

"Proponents of the Comprehensive Legislative Model contend that a public key infrastructure, such as that set forth in this model, is the most effective means for facilitating electronic commerce, and that, due to liability concerns, commercial CA's will not enter the market until legislation defining the rights and liabilities of electronic commerce participants is put in place.

However, critics of the Comprehensive Legislative Model contend that this model will not facilitate electronic commerce for the following reasons:" [McC97]:

1. The electronic marketplace is still in its infancy and it is therefore premature to enact comprehensive legislation;

2. Legislation should take place in order to take care of identifiable problems, after the industry has matured;

3. The marketplace, not legislation, should dictate the direction of the electronic commerce industry;

4. By enacting a Comprehensive Legislative Model, the market could be skewed, facilitating a business model for which there is no demand or place in the marketplace;

5. The liability concerns expressed by proponents of the Comprehensive Legislative Model are the product of flawed business models (i.e., the Open PKI Model)[1] not flawed law, and are not keeping commercial CA's out of the marketplace, as evidenced by the number of CA's entering the marketplace in the absence of legislation;

6. Legislation should not be technology-specific, as other forms of electronic authentication may be just as, or more suitable, for authenticating documents as public key cryptography, and

7. Legislation should not override all writing requirements, as an electronic record may not be sufficient in all circumstances.

4.2 Dutch national TTP project

"The Dutch national TTP project is based on the principles of market forces and deregulation. Among other things this implies that the development of a TTP infrastructure is regarded as primarily a market responsibility" [N.V98].

In retrospect of section 4.1 this would define the TTP project a Minimalist Legislative approach.

The National TTP project has three aims:

1. The formulation of requirements for the provision and use of TTP services;

2. Compiling a survey of instruments with which these requirements can be safeguarded;

3. Encouraging the development of a Dutch TTP infrastructure.

"When it comes to the storage of data and exchange of messages, confidence and security are consequently becoming ever more important. An important means of safeguarding these aspects is the use of Trusted Third Parties (TTP's), which together form a TTP infrastructure" [N.V98].

[1] "Critics contend that the Open PKI Model is not a winning business model because it involves considerable liability risks and the costs associated with its implementation cannot be internalized. Moreover, say critics, Comprehensive Legislative Models attempt to solve the liability problem by shifting many of the liability risks to consumers by (i) not putting any caps on the liability of a subscriber thereby exposing the subscriber to unlimited liability, (ii) putting a cap on the liability of CA's which could unfairly result in unrecompensed damages of third parties, and it (iii) imposing evidentiary presumptions that are burdensome on the subscriber. In fact, it has been pointed out, a subscriber faces greater liability in certain situations, such as, in the event of a forgery, than it would face in analogous situations, such as credit card transactions, where its liability would be limited" [McC97].

Since TTP's have become the focus of international attention, the Dutch government decided in early 1997 to initiate a national TTP project. The Dutch national TTP project relates solely to *public* TTP services, defined as:

> *"Services that are in principle available to any individual, business or institution and/or are offered via a public infrastructure" [N.V98].*

The Dutch national TTP project is based on two underlying principles, namely that of *market forces* and *deregulation.* In this respect the development of a TTP infrastructure is regarded as primarily the responsibility of the market.

4.2.1 TTP services

The definition of a trusted third party in the Dutch definition is:

> *"Trusted Third Parties (TTP's) are organizations that provide services in order to enhance the reliability of electronic data exchange"* [N.V98].

Reliability comprises the following quality aspects, in the point of view, of the TTP project:

- the *authenticity* of data;
- the *integrity*, or the accuracy and completeness of data;
- the *reliability* of data[2].

In the view of the Dutch TTP project there is a fundamental difference between TTP services aimed at protecting the authenticity and/or integrity of data, messages and transactions and TTP services aimed at protecting the confidentiality of data, messages and transactions:

1. *TTP services to protect authenticity and integrity*
   TTP services to protect authenticity and integrity include the provision of digital certificates (where the TTP fulfills the role of Certification Authority (CA)); the placement and verification of digital signatures; irrefutable confirmation of transmission and receipt of electronic messages (non-repudiation); the management of cryptographic keys to protect authenticity and integrity, with the exception of the storage of private keys; and the time-stamping of electronic reports.
2. *TTP services to protect confidentiality*
   TTP services to protect confidentiality include the encryption of electronic communications and the management of cryptographic keys for confidentiality.

[2]Reliability in this sense strongly rest on the services provided by the TTP. The services provided by these TTP's are in many cases based on the use of cryptographic techniques. Among other things they relate to the provision of electronic certificates; the placement and verification of digital signatures; the encryption of electronic communications; the generation, distribution, storage and/or destruction of cryptographic keys (i.e. key management); non-repudiation; and the saving and time-stamping of electronic messages and data, in encrypted form or otherwise.

4.2.2 Requirements with respect to TTP services to protect authenticity and integrity

TTP services to protect authenticity and integrity include:

1. The provision of digital certificates;
2. The placement and verification of digital signatures;
3. Irrefutable confirmation of transmission and receipt of electronic messages (non-repudiation);
4. The management of cryptographic keys to protect authenticity and integrity, with the exception of the storage of private keys; and
5. The time-stamping of electronic messages.

Within TTP services to protect authenticity and integrity a distinction is often drawn between the Registration Authority (RA), with responsibility for the legitimation of affiliated users, and the Certification Authority (CA), which provides electronic certificates on behalf of those users.

The requirements applying to this category of TTP's are the subject of a certain degree of consensus among the parties concerned. Among other things the requirements relate to the legal status of digital signatures, the reliability of the TTP service provided and the TTP itself, the protection of privacy, and international interoperability [N.V98].

These issues will be discussed in the following subsections.

4.2.2.1 Legal status of digital signatures

The legal status of digital signatures is currently subject of research of the Dutch national TTP project. At international level policies are being worked out by United Nations Commission On International Trade Law or UNICITRAL [oEC98]. At the European level, a communication has been issued announcing that the EU is seeking to introduce an European directive with respect to the mutual recognition of digital signatures.

At European level a distinction is made between the TTP organization and the TTP service.

4.2.2.2 Reliability of the TTP organization

The following reliability criteria apply to the TTP [N.V98]:

- *lawfulness* - TTP's must act in accordance with national and international law in every sense of the word;
- *financial position* - the financial position of the TTP organization must provide sufficient guarantees with respect to the continuity of the TTP service;

- *business continuity* - the continuity of TTP services must be guaranteed as far as possible, e.g. in the event of a takeover, merger, strike or bankruptcy;
- *security* - the confidentiality, integrity and availability of data and information systems within the TTP organization must as far as possible be protected by means of an adequate system of measures against loss arising from catastrophes, breakdowns and intentional and unintentional human actions. The Code of Practice for Information Security Management (British Standard 7799) would appear to provide a good basis for the security of TTP organizations, although supplementary measures may be required in the light of the stringent reliability requirements that a TTP must satisfy. The Information Technology Security Evaluation Criteria (ITSEC) and Common Criteria based on the ITSEC may provide a good basis for the evaluation of the IT-products used;
- *personnel* - the owners, shareholders, directors, management and personnel of the TTP organization must command trust;
- *authentication* - with respect to the taking of decisions and the performance of activities by its management and staff, the TTP organization must at all times be able unambiguously to establish the identity of the individuals in question. Determining the identity of the individuals in question should also be possible in retrospect;
- *authorization* - specific authorizations need to be assigned clearly and unambiguously to specific positions and officers within the TTP organization;
- *separation of duties* - there must be a sufficient audit separation between managerial, custodial, executive and control functions within the TTP organization, e.g. in relation to key management;
- *supervision* - in order to guarantee the reliability of a TTP organization regular checks will need to be made by an independent body to establish whether the organization is complying with a previously drawn up package of criteria and requirements and whether that package is sufficient to achieve the desired degree of reliability. Various models for achieving such a form of supervision are conceivable. It may also be emphasized once again that the relevant legal requirements with respect to supervision and control also apply to TTP organizations supplying services to protect specific public functions;
- *carefulness* - the TTP must solely supply data to third parties where there is a demonstrable legal basis for doing so. It is highly important that the users of a TTP service can rely on the fact that any provision of data will take place under strict conditions only and then only if there is a demonstrable statutory basis, in which respect the TTP organization will need to convince itself of the legitimacy of any request for cooperation. Private keys used solely to protect authenticity and integrity must in no circumstances be provided to third parties;
- *management of operating assets* - the development and management of information technology and other operating assets within the TTP should be arranged in accordance with generally accepted quality standards;

- *independence* - the TTP must not be tied to one or more existing parties and must not have an interest in any information to be protected;
- *transparency* - the TTP must provide insight into its working methods to permit appraisal of the TTP organization and its services.

4.2.2.3 Reliability of the TTP service

The following reliability criteria apply to a TTP service [N.V98]:

- *reliable technology* - the technology must be sufficiently reliable to ensure the confidentiality, integrity and availability of automated data processing;
- *documentation* - the design, implementation, management and use of the TTP service must be adequately documented;
- *key management* - key management must be reliable.

A large number of national and international parties are involved in the development and use of TTP infrastructures. In this respect a basic distinction may be drawn between market players and government authorities. A special situation applies however to parties with a statutory authorization to obtain electronic data; this applies to both market players and government authorities.

The most important role is set aside for the market players, or the suppliers and users of TTP services. TTP services are provided by a very wide range of commercial and non-commercial organizations, which may or may not operate within a specific market segment. The users of TTP services include businesses, institutions and private individuals.

Given the public issues at stake, a role for the government is set aside in the form of policy, incentives, legislation and regulations and/or supervision. The general notion in this regard is that the government, making use of the available instruments, at least has a task to protect the public and individual citizens, e.g. by ensuring the reliability of TTP services, promoting national and international interoperability, protecting the privacy of the users of a TTP service and safeguarding lawful access to electronic data. In addition, the government can encourage the development of a safe and reliable TTP infrastructure. Finally the government can operate as a market player - as a customer but also as a provider of TTP services.

4.3 German legislation

As one of the first countries, Germany developed comprehensive legislation for the use of digital signatures [FMoET97] [Sig97]. In retrospect of section 4.1, this would define the German law as a Comprehensive Legislative approach. The purpose of the Act is as follows:

> *"The purpose of this Act is to establish general conditions under which digital signatures are deemed secure and forgeries of digital signatures or manipulation of signed data can be reliably ascertained"* [FMoET97].

The German law thus does not explicitly state that digital signatures which can be verified with a certificate that has been issued by a CA are legally rightful [Dut98].

In the German Law, "the operation of a certification authority shall require a license from the competent authority" [FMoET97].

"A license shall be denied when facts warrant the assumption that the applicant does not possess the reliability necessary to operate a certification authority, when the applicant does not furnish proof of the specialized knowledge required to operate a certification authority.

The required specialized knowledge shall be deemed available when the persons engaged in the operation of the certification authority have the necessary knowledge, experience and skills. The other requirements pertaining to the operation of the certification authority shall be deemed met when the competent authority has been notified in a timely manner by means of a security concept of the measures ensuring compliance with the security requirements in this Act" [FMoET97].

For digital signatures a safeguard catalogues has been drafted by the German Information Security Agency, or BSI [Age97]. The catalogue describes how the individual technical components and the organizational environment are to be configured and structured in order to achieve an overall system in which digital signatures can be created which possess the necessary degree of security to prevent forgery and manipulation.

The security concept consists of a 'complete' overview of all measure taken, to insure 'every' possible problem has been securly dealt with. These include [Age97]:

- General security requirements and security policy.
- Functional security requirements for the CA.
- Security requirements and recommendations for the registration authority
- Security requirements and recommendations for revocation management.
- Security requirements and recommendations for security concept and documentation.
- Security requirements and recommendations for the organizational structure
- Security requirements and recommendations for the personnel.
- Security requirements and recommendations for the infrastructure.
- Security requirements and recommendations on IT

These security requirements all take place at the TTP and should insure that certificates are correct and valid. Thus these requirements relate back to the authenticity aspect, as stated in section 2.1 of chapter 2.

4.3.1 Criticism of German Law

The German law takes an hierarchical approach to a TTP as can be seen in chapter 3 figure 3.5.a. In the German model, "the authority shall issue the certificates for the signature keys used for affixing signatures to certificates. The authority shall keep

the certificates which it has issued available for verification and retrieval at all times and for everyone over publicly available telecommunication links" [FMoET97]. If the private key of the *"root-CA"* is ever compromised, then all certificate signed with this key are compromised.

The German law is also very rigid. Every CA needs a separate license and a certificate of the central "root-CA". It is not possible for a infrastructure to get one license and then "design" their own hierarchy.

The German Law also only recognizes CA's that do "everything". This limits the organizational possibility to divide certain functionality among different organizations [Dut98].

4.4 Utah digital signature act

The Utah digital signature act has been made to, facilitate commerce by means of reliable electronic messages, to minimize the incidence of forged digital signatures and fraud in electronic commerce, to implement legally the general import of relevant standards and to establish uniform rules regarding the authentication and reliability of electronic messages.

The Utah act does not force a CA to get a license, but certificates issued by a non-licensed CA, do not have any legal value. The Utah Digital Signature Act does not have a mandatory licensing obligation but a indirect or implicit one.

The Utah act uses a *"Root"-Certification Authority* called *the division.* The division is a certification authority, and may issue, suspend, and revoke certificates. The division further maintains a publicly accessible database containing a certification authority disclosure record for each licensed certification authority and publishes the contents of the database in at least one recognized repository. The devision further makes rules including:

- governing licensed certification authorities, their practice, and the termination of a certification authority's practice;
- determining an amount appropriate for a suitable guarantee, in light of:
  - the burden a suitable guarantee places upon licensed certification authorities; and
  - the assurance of financial responsibility it provides to persons who rely on certificates issued by licensed certification authorities;
- for reviewing software for use in creating digital signatures and publish reports concerning software;
- specifying reasonable requirements for the form of certificates issued by licensed certification authorities, in accordance with generally accepted standards for digital signature certificates;
- specifying reasonable requirements for record keeping by licensed certification authorities;

- specifying reasonable requirements for the content, form, and sources of information in certification authority disclosure records, the updating and timeliness of such information, and other practices and policies relating to certification authority disclosure records; and
- specifying the form of certification practice statements.

To obtain or retain a license a certification authority shall:

- be the subscriber of a certificate published in a recognized repository;
- employ as operative personnel only persons who have not been convicted of a felony or a crime involving fraud, false statement, or deception;
- employ as operative personnel only persons who have demonstrated knowledge and proficiency;
- file with the division a suitable guarantee, unless the certification authority is the governor, a department or division of state government, the attorney general, state auditor, state treasurer, the judicial council, a city, a county, or the Legislature or its staff offices provided that:
  1. each of the above named governmental entities may act through designated officials authorized by ordinance, rule, or statute to perform certification authority functions; and
  2. one of the above-named governmental entities is the subscriber of all certificates issued by the certification authority;
- have the right to use a trustworthy system, including a secure means for controlling usage of its private key;
- present proof to the division of having working capital reasonably sufficient, according to rules of the division, to enable the applicant to conduct business as a certification authority;
- maintain an office in Utah or have established a registered agent for service of process in Utah; and
- comply with all other licensing requirements established by division rule.

4.4.1 Legal status of digital signature

Where a rule of law requires a signature, or provides for certain consequences in the absence of a signature, that rule is satisfied by a digital signature if:

- that digital signature is verified by reference to the public key listed in a valid certificate issued by a licensed certification authority;
- that digital signature was affixed by the signer with the intention of signing the message; and

- the recipient has no knowledge or notice that the signer either:
  - breached a duty as a subscriber; or
  - does not rightfully hold the private key used to affix the digital signature.

A message is as valid, enforceable, and effective as if it had been written on paper, if it:

- bears in its entirety a digital signature; and
- that digital signature is verified by the public key listed in a certificate which:
  - was issued by a licensed certification authority; and
  - was valid at the time the digital signature was created.

There is some criticism concerning the Utah act. This concerns the criteria that a CA has to meet in order to get a license. Only a few CA's can satisfy all requirements and thus can receive a license. Only licensed CA's adhere to the digital signature act and thus submit legal signatures [Dut98].

4.4.2 Overview of regulation

In table 4.1 a overview follows of the nature and purpose of several drafts and laws [Dut98].

	The Netherlands	*Germany*	*Utah*
purpose regulation	negative, licensing is voluntary	positive, licensing is mandatory	negative, licensing is voluntary
nature regulation	facilitate electronic commerce	technical framework for the use of digital signatures	legalisation of digital signatures

Table 4.1: Overview of the nature and purpose of several drafts and laws

4.5 Crypto regulation

> *"Governments have long restricted export of cryptography for fear that their intelligence activities are hampered by the crypto use of foreign states and scoundrels. Since the rise of crypto use over the past decades, governments increasingly worry about criminals using cryptography to thwart law enforcement. Thus, many countries are considering laws focusing on maintaining law-enforcement and national-security capabilities through regulation of cryptography"* [Koo98].

"Criminals and terrorists may take advantage of the concealing merits of cryptography to remain out of reach from wiretapping officials. It is because of this nefarious

use of cryptography that governments have long restricted its export, and are now also considering to regulate its use domestically" [Koo97].

The other side of the story is the fact that cryptographic mechanism are an enormously important tool for information security and thus governments are stimulating its use.

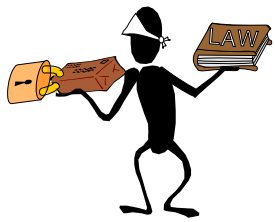


Figure 4.1: Balance between secure transfer and crypto regulation

4.5.1 A brief survey of crypto regulation

The export regulations of cryptography in Europe are harmonized by both an EU decision, from December 1994 on dual-use goods, and the Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies. Both regulate the export of dual-use goods; cryptography is such a dual-use good, as it has both military and civil applications. The general drift of these regulations is that you need a license to export crypto hardware and software, with the exception of mass-market and public-domain software. Export within the EU should be easier, although some manufacturers complain that here also, bureaucratic procedures have to be followed. Most countries in the EU and a host of other, mainly developed, countries have recently implemented the Wassenaar Arrangement [Koo97].

In Europe, only France and Russia have considerable restrictions, but lately, some other countries have also stated intentions to regulate crypto. The laws in France and Russia are a near complete prohibition on crypto use, sale and manufacture.

Other attempts to control cryptography have taken place in various countries. Belgium adopted a law in late 1994 which was noticed only in 1996 to contain a provision that might be interpreted as a prohibition of using cryptography in telecommunications. If you use 'equipment which renders tapping ineffective', your telecommunications equipment might be seized, according to the provision. Some have seen this as a requirement to use escrowed encryption[3]. One member of parliament has proposed dropping the provision and instead requiring people to decrypt if this is necessary for the investigation [Koo98]

[3]Key escrow is a capability that allows authorized persons or agencies, under certain prescribed conditions, to read the keys used with the help of information supplied by one or more trusted parties storing escrowed parts of the used keys [Poh97].

4.5.1.1 Export Control

COCOM, the Coordinating Committee for Multilateral Export Controls, was an international organization for the mutual control of the export of strategic products and technical data from country members to prescribed destinations. It maintained, among others, the International Industrial List and the International Munitions List.

The main goal of the COCOM regulations was to prevent cryptography from being exported to "dangerous" countries - usually, the countries thought to maintain friendly ties with terrorist organizations, such as Libya, Iraq, Iran, and North Korea. Exporting to other countries is usually allowed, although states often require a license to be granted.

In 1995, 28 countries decided to establish a follow-up to COCOM, the Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies. The negotiations on the treaty were finished in July 1996, and the agreement was signed by 31 countries[4].

The Wassenaar Arrangement controls the export of weapons and of dual-use goods, that is, goods that can be used both for a military and for a civil purpose; cryptography is such a dual-use good. The provisions are largely the same as COCOM regulations. The General Software Note excepts mass-market and public-domain crypto software from the controls; five countries, including the US and the UK, deviate from the GSN and control the export of mass-market and public-domain crypto software [Koo98].

Apart from the restriction of cryptography, another part of legislation is aimed at accessing the contents of cryptographicly enciphered messages. This can be achieved if one has access to the cryptographic key material.

4.5.1.2 Key-escrow

As part of a possible solution, for governments and their needs to have access to information to enforce the law, to protect the national security, a great many countries are focusing on the use of key-escrow. In the view of several governments they had three options [Kam94]:

- To do nothing, resulting in the possible proliferation of products with encryption capabilities that would seriously weaken, if not wholly negate, the authority to wiretap and damage intelligence collection for national security and foreign policy reasons;
- To support an approach based on weak encryption, likely resulting in poor security and cryptographic confidentiality for important personal and business information; and
- To support an approach based on strong but escrowed encryption. If widely adopted and properly implemented, escrowed encryption could provide legitimate users with high degrees of assurance that their sensitive information would

[4]These include: Argentina, Australia, Austria, Belgium, Canada, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Japan, Luxembourg, the Netherlands, New Zealand, Norway, Poland, Portugal, the Republic of Korea, Romania, the Russian Federation, the Slovak Republic, Spain, Sweden, Switzerland, Turkey, the United Kingdom, the United States, Bulgaria and Ukraine.

> remain secure but nevertheless enable law enforcement and national security authorities to obtain access to escrow-encrypted data in specific instances when authorized under law. Moreover, the Administration hoped that by meeting legitimate demands for better information security, escrowed encryption would dampen the market for unescrowed encryption products that would deny access to law enforcement and national security authorities even when they sought access for legitimate and lawfully authorized purposes.

As part of the solution the US government introduced the Clipper Initiative. The Clipper initiative was conceived as a way for providing legal access by law enforcement authorities to encrypted telephony.

4.5.1.3 The Clipper initiative

The Clipper chips is an integrated circuit chips that is build into devices used for voice communications. These chips are part of an overall system and provide voice confidentiality for the user and exceptional access to law enforcement authorities. To provide these functions, the Clipper chip was designed with a number of essential characteristics [DL96]:

- Confidentiality would be provided by a classified algorithm known as Skipjack. Using an 80-bit key, the Skipjack algorithm would offer considerably more protection against brute-force attacks than the 56-bit DES algorithm. The Skipjack algorithm was reviewed by several independent experts, all with the necessary security clearances. In the course of an investigation limited by time and resources, they reported that they did not find short-cuts that would significantly reduce the time to perform a cryptanalytic attack below what would be required by brute force.

- The chip would be protected against reverse engineering and other attempts to access its technical details.

- The chip would be factory-programmed with a chip-unique secret key, the ”unit key” or ”device key,” at the time of fabrication. Possession of this key would enable one to decrypt all communications sent to and from the telephone unit in which the chip was integrated.

- A law enforcement access field (LEAF) would be a required part of every transmission and would be generated by the chip. The LEAF would contain two items:

  - the current session key, encrypted with a combination of the device-unique unit key, and
  - the chip serial number.

  The entire LEAF would itself be encrypted by a different but secret ”family key” also permanently embedded in the chip. The family key would be the same in all Clipper chips produced by a given manufacturer.

To manage the use of the LEAF, the U.S. government would undertake a number of actions:

- The unit key, known at the time of manufacture and unchangeable for the life of chip, would be divided into two components, each of which would be deposited with and held under high security by two trusted government escrow agents located within the Departments of Commerce and Treasury.
- These escrow agents would serve as repositories for all such materials, releasing the relevant information to law enforcement authorities upon presentation of the unit identification and lawfully obtained court orders.

When law enforcement officials encountered a Clipper encrypted conversation on a wiretap, they would use the LEAF to obtain the serial number of the Clipper chip performing the encryption and the encrypted session key. Upon presentation of the serial number and court authorization for the wiretap to the escrow agents, law enforcement officials could then obtain the proper unit-key components, combine them, recover the session key, and eventually decrypt the encrypted voice communications.

As a Federal Information Processing Standards Publication (FIPS), the Escrowed Encryption Standard (EES) is intended for use by the federal government and has no legal standing outside the federal government. Private consumers are free to decide whether or not to use EES-compliant devices to safeguard communications and are free to use other approaches to communications security should they so desire [DL96].

Another approach adhere to the law, is to make a distinction between confidentiality and authentication of data. Thus a message could be signed with a strong key but sent unencrypted. Thus the message could be read by anyone but not altered because this would be noticed due to the digital signature attached. This approach would however meet the requirements a many users that need confidentiality.

4.6 Summary

At the start of this chapter it was shown how crypto regulation is making an effort to provide people with a robust, reliable crypto system. Legislation is put in place to assure commerce conducted with the help of the Internet has a secure basis. Standards are generated and implemented to assure the public is provided with secure and tested cryptography. The end-product should be, a product that is secure and “trusted” by all users.

Countries take different approaches to how legislations should be implemented. Their is however broad concensus as to how a TTP should implement its organization. These include, as stated in section 4.2.2.2, such aspect as lawfulness, financial position of the TTP, business continuity, security, personnel, authentication, authorization, separation of duties, supervision, carefulness, management of operating assets, independence and transparancy.

At the other end, governments are raising concerns whether law enforcement is being obstructed by the proliferation of unbridled growth and usage of cryptography. Measure are taken to insure encrypted messages can be read by law enforcement.

These range from the complete ban of encryption[5], a ban on "strong encryption"[6] and most of all the use of key-escrow agents[7].

Providing a secure and robust crypto system and giving governments access to this information, are two different aspects that do not mix. There are several reasons for this [Koo98]:

- Crooks and criminals have never been law-abiding and will simple use other means of cryptography;
- A second obstacle to key recovery is liability. Agencies that provide law enforcement with access to keys, either through key deposits or through sent-along session keys, will be a target for attacks, and loss of keys may have huge financial consequences. Therefore, it is to be expected that key recovery agencies will want to exonerate themselves from liability, at least to a large extent. Will users accept this?
- The third issue in key recovery is constitutional rights. At stake is the right to privacy, including the right to confidential communication, a right established by human rights covenants and many constitutions. If states implement key recovery, they must see to it that this constitutional right is not hampered. Although potentially, key recovery is compatible with the right to privacy, when you start implementing key recovery, you will have to address the extra risk such a system poses to privacy - after all, key recovery systems are inherently weaker than non-recovery systems.
- Although most governments recognize that crypto policies cannot be implemented on a purely national basis, the international discussions indicate that it is difficult indeed to establish cooperation between states. For instance, key recovery depends on access to keys, and for this to work on any scale, international access to keys must be safeguarded somehow[8].
- Key recovery schemes have been proposed only since 1993, and consequently, have not been researched to the extent that traditional systems as RSA or DES have. It is a primary cryptographic principle that the strength of a system can not be proven theoretically, but has to be proven in practice, by long years of attacks by cryptanalysts. Key recovery systems have yet to be subjected to thorough examinations by the cryptographic community.

One can conclude that legislation is not uniform across different nations. Some countries take an active approach, that often is very rigid and other lean back and wait what will come. There is however a broad consensus that some form of international legislation has to be created that will give information sent across the internet a legal status. This may however take some time.

[5]The laws in France and Russia are a near complete prohibition on crypto use, sale and manufacture [Koo98]

[6]The US export control allows a maximum of 40-bit symmetric crypto being exportable

[7]The UK has been leaning towards depositing of keys with TTP's as have the US [Koo98].

[8]Suppose that the US, UK, France and Russia will implement key recovery systems: how will they handle communications with the rest of the world?

Chapter 5

Specific aspects of the PKI model

In the previous chapters an outlay has been given of the PKI model and the actors involved. It has been explained, in chapter 2, how information can be protected with the help of encryption. A model has been given how to protect information with the help of public key encryption. Chapter 3 has given a model on how information can be sent across networks in a safe manner and comply with the requirements that have been formulated. Chapter 4 has given an insight into the requirements that have been stated by legislators.

Certain aspects of digital signatures have been given but no further explanation was given as to how these should provide adequate security. This concerns the following:

- aspects of keys and more specific how good keys can be generated, how they should be stored and deleted;
- aspects of certificates. This will include a general history and current status of the "official" standard, X-509;
- aspects of public key encryption methods and different standards such as RSA, DSS etc ....
- aspects of trap-door and/or hash-functions, what standards are used and what can be said of these standards.

These issues will be explained more thoroughly in this chapter. Aspects of this chapter will perhaps become obsolete within a year, but currently these are very actual and give a better understanding of the issues involved.

5.1 Aspects of keys

Encryption today is done with the use of a well known algorithm and secret or private key. If the key is known by an intruder then this can have consequences. This would enable the attacker both to read all messages encrypted with the public key and to forge signatures. Information concerning the key could be gained at several points:

- at the key generation, if someone could have information how a key is generated;

- when a message is sent, if someone could decipher the information with an attack on the message itself. This has been explained in chapter 2[1].
- the storage of the key, if the key could be stolen from a person then also a problem arises.

5.1.1 Key generation

> *"Security systems today are built on increasingly strong cryptographic algorithms that foil pattern analysis attempts. However, the security of these systems is dependent on generating secret quantities for passwords, cryptographic keys, and similar quantities. The use of pseudo-random processes to generate secret quantities can result in pseudo-security. The sophisticated attacker of these security systems may find it easier to reproduce the environment that produced the secret quantities, searching the resulting small set of possibilities, than to locate the quantities in the whole of the number space" [DE94].*

When a random number generator is used in the key generation process, all values must be generated randomly or pseudo-randomly[2] in such a manner that all possible combinations of bits and all possible values are equally likely to be generated [FIP94].

5.1.1.1 Deciphering a key

In most cases, an adversary can try to determine the "secret" key by trial and error. This is possible as long as the key is enough smaller than the message that the correct key can be uniquely identified. The probability of an adversary succeeding at this must be made acceptably low, depending on the particular application. The size of the space the adversary must search is related to the amount of key "information" present in the information theoretic sense [Sha63]. This depends on the number of different secret values possible and the probability of each value as follows:

$$Bits - of - info = \sum_i -p_i * log_2(p_i) \tag{5.1}$$

where i varies from 1 to the number of possible secret values and p_i is the probability of the value numbered i. Since p_i is less than one, the log will be negative so each term in the sum will be non-negative.

If there are 2^n different values of equal probability, then n bits of information are present and an adversary would, on the average, have to try half of the values, or $2^{(n-1)}$, before guessing the secret quantity. If the probabilities of different values are unequal, then there is less information present and fewer guesses will, on average, be

[1] Because the algorithm is believed to be secure this will not be dealt with here.

[2] Statistician have since long made use of random numbers to test theories. These random number generators are actually pseudo-random numbers, generated in a deterministic way, which only appear to be random in a *statistical* sense.

required by an adversary. In particular, any values that the adversary can know are impossible, or are of low probability, can be initially ignored by an adversary, who will search through the more probable values first [DE94].

For example, consider a cryptographic system that uses 56 bit keys. If these 56 bit keys are derived by using a fixed pseudo-random number generator that is seeded with an 8 bit seed, then an adversary needs to search through only 256 keys, by running the pseudo-random number generator with every possible seed, not the 2^{56} keys that may at first appear to be the case. Only 8 bits of "information" are in these 56 bit keys.

5.1.1.2 Finding random sources

What is a truly random number? The definition can get a bit philosophical. Knuth speaks of:

> *"a sequence of independent random numbers with a specified distribution, each number being obtained by chance and not influenced by the other numbers in the sequence" [Knu81].*

Rolling a die would give such results. But computers are logical and deterministic by nature, and fulfilling Knuth's requirements is not something they were designed to do [Mat96]. This is why it is believed a computer, without some random source, cannot produce truly random numbers.

For the present, the lack of generally available facilities for generating such unpredictable numbers is an open wound in the design of cryptographic software. For the software developer who wants to build a key or password generation procedure that runs on a wide range of hardware, the only safe strategy so far has been to force the local installation to supply a suitable routine to generate random numbers. To say the least, this is an awkward, error-prone and unpalatable solution [DE94].

For the generation of cryptographic key's, methods are required that give no information as to what the key might look like. Most traditional sources of random numbers use deterministic sources of "pseudo-random" numbers. These typically start with a "seed" quantity and use numeric or logical operations to produce a sequence of values. These state values, are then sent through a hash-function. The strength of this approach relies on a hash-function being a one-way function, from the random output bytes it is difficult to determine the state value, and hence the other output bytes remain secure.

If the attackers cannot guess or predict the seeds, they will be unable to predict the output. There are two aspects to a random seed: quantity and quality. They are related. The quality of a random seed refers to the entropy[3] of its bits.

There are several ways to provide "random" seeds. Some methods to obtain random seeds produce "less" randomness then one would perhaps think. Ways to obtain seeds are shown in table 5.1.

[3] In a system that produces the same output each time, each bit is fixed, so there is no uncertainty, or zero entropy per bit. If every possible sequence of outputs is equally likely, i.e. truly random, then there is maximum uncertainty, or one bit of entropy per output bit [Mat96]

System Unique	*Variable and Unguessable*	*External Random*
Configuration files	Contents of screen	Cursor position with time
Drive configuration	Date and time	Keystroke timing
Environment strings	High resolution clock	Microphone input (with samples microphone connected)
	Last key pressed	Mouse click timing
	Log file blocks	Mouse movement
	Memory statistics	Video input
	Network statistics	
	Process statistics	
	Program counter for other processes or threads	
Less Entropy ←	→ *More Entropy*	

Table 5.1: Seed sources [Mat96]

The problem with the random numbers generated in the method above is that they are not "truly" random. This is why German law does not accepts keys to be generated in the way mentioned above [Sch98].

Generating truly strong random numbers is actually quite easy. All that's needed is a physical source of unpredictable numbers. A thermal noise or radioactive decay source and a fast, free-running oscillator would do the trick directly [Gif98]. This is a trivial amount of hardware, and could easily be included as a standard part of a computer system's architecture. All that's needed is the common perception among computer vendors that this small additional hardware and the software to access it is necessary and useful [DE94].

Many computers come with hardware that can, with care, be used to generate truly random quantities. Increasingly computers are being built with inputs that digitize some real world analog source, such as sound from a microphone or video input from a camera. Under appropriate circumstances, such input can provide reasonably high quality random bits.

5.1.2 Key storage

After a key has been generated another weak link in the PKI-model concerns the storage of the private key. If the key can be comprised with or without the user knowing this, then an attacker can presume the identity of the owner. Depending on the the time-lag between a compromised key and the user revoking his key, this can have serious consequences.

To insure an attacker cannot obtain or use the private key of an entity without his knowledge, several measures have to be taken:

- *Copying* of a private key should not be possible, without the owning entity finding out, this would insure the private key from being copied without the owning

entities knowledge;

- *Use* of a private key should not be possible, without some private knowledge, this would insure a stolen key from being used by an attacker.

To insure a private key from being copied, it has to be assured the private key cannot be read. Reading a key means being able to copy it. Some sort of measure should be taken to insure a key from being read. A possible measure could be to store the private key in some sort of black box or token. A box that would take as input a plaintext and produce a ciphertext as output. German law and other standards require the private keys to be stored in cryptographic modules in such way that they can be used inside the token but never be retrieved from the token [Sig97] [SEI98].

Should the private key somehow be compromised, then a measure has to be taken to insure it can then not be used without some special knowledge. This could be achieved with the help of a *personal identification number* or *PIN*. By storing the private key of the entity in encrypted format, with a PIN-code as the key, it can be secured against unauthorized usage. The strength of this method however depends on the length of the key[4], not on the amount of tries.

If the private key is imbedded in a token then the amount of tries to crack the private key can be limited. This should be limited to three [SEI98].

5.2 Aspects of certificates

Because of the international aspects of commerce, certificates have to be standardized. A standard that has received worldwide acknowledgment and is used worldwide concerns the *Authentication Framework of the ITU-T* or *ITU-T Recommendation X-509* standard [X.597].

5.2.1 History X-509

> *"Imagine having a telephone in your house, but not having a telephone directory, or recourse to the 'directory enquiries' service. The phone now becomes a lot less useful than it was. How can you telephone your Auntie Margery in Australia to wish her happy birthday? You may not remember which digits you need to dial for international access, nor can you remember all the several hundred different country codes. And what if she has recently moved house, but you don't yet have her new address? Or what if you want to ring up several stores in town to see which one of them has the latest 'green' garden fertiliser in stock, and which one has the most competitive price?"* [Cha96]

"The examples might be flippant, but they make the point. While it may be possible to keep your own address book of friends, colleagues and relatives that you call most frequently, it certainly is not possible to keep an address book of everyone whom you have ever called, or whom you are likely to want to call in the future. And who is

[4] Cracking a key can be done in several ways as can be seen in Appendix A.

going to update your personal directory when people change addresses, or get a new job, or install a new line for a fax machine?" [Cha96].

While the telephone companies have trouble keeping up with managing all the telephone and fax numbers, keeping track of Internet addresses poses an even greater problem. This lead to the development of a directory that would keep track of all these changes.

In 1984, CCITT[5] drafted its X.400 recommendation, whose major concern was to provide a white pages service that would return either the telephone numbers or X.400 O/R addresses[6] of people. On the other side there was the ISO[7] and the ECMA[8], who were concerned mainly with providing the name server service for Open Systems Interconnection, OSI, applications. The two tracks merged in 1986, with the formation of the Joint ISO/CCITT working group on Directories.

The X.500 Standard assumes distributed administration of the database, and specific functions are built in for this.

Because the X.500 directory contains information that should not be available to everyone, access security was added. This became the X-509 recommendation or Authentication Framework.

5.2.2 Usage of the X-509

Distinguished names are the standard form of naming in an ITU-T X.500 directory and in X-509 certificates. Distinguished names were intended to identify entities in the X.500 directory tree. A relative distinguished name is the path from one node to a subordinate node. The entire distinguished name traverses a path from the root of the tree to an end node that represents a particular entity. A goal of the directory was to provide an infrastructure to uniquely name every communications entity everywhere, hence the "distinguished" in "distinguished name". As a result of the directorys goals, names in X-509 certificates are perhaps more complex than one might like, e.g., compared to an e-mail address. Nevertheless, for business applications, distinguished names are worth the complexity, as they are closely coupled with legal name registration procedures, something that simple names such as e-mail addresses do not offer [RSA96].

Nevertheless the X.500 has not received broad acceptance in the Internet community. As it stands the most used system for naming on the Internet is based on Simple Mail Transfer Protocol (SMTP).

[5]Comité Consultatif International Téléphonique

[6]The X.400 address describes a format with which it is possible to access specific information or a specific person. An X.400 address can contain a number of different items such as: name, common name, locality name, state name, organization name, organizational unit, title, match, email address, country name domain name, given name, initials, numeric user id, organizational unit name, organizational units, postal code, surname, terminal id, address. An X.400 address looks like: */C=US/SP=MASSACHUSETTS/L=CAMBRIDGE/PA=360 MEMORIAL DR./CN=KEN SMITH/*

[7]International Standards Organisation

[8]European Computer Manufacturers Association

5.2.3 Information in a X-509 certificate

The initial version of X-509 was published in 1988, version 2 was published in 1993, and version 3 was proposed in 1994 and considered for approval in 1995. Thus the current standard which is often referred to is the X-509v3.

The differences concerning version 1, 2 and 3 include identifiers and extensions that were added to the standard. These were added to give more meaning to the role and/or name of the signer. When signing a contract the role of the signer, e.g. Financial Director from the Delta Company, is more relevant than its name. Also, at the moment, no naming scheme is specified for digital signatures. That leads to the following key question: which technique should be used to point unambiguously to a person, or entity, that can be easily recognized and traced [(SE98].

The X-509 certificate has the content as given in table 5.2.3 [X.597]. These contents should make it possible to uniquely identify the object.

Name of the field	*Contents and functions*
Version	The version number of the certificate, e.g. 1,2 or 3
Serial number	Unique identifier for each certificate generated by issuer; integer
Signature	Algorithm identifier and algorithm used to sign certificate
Issuer	Name of issuer, certificates may employ a variety of name forms, including Internet electronic mail names, Internet domain names, X.400 originator/recipient addresses, and EDI party names.
Validity	NotBefore and NotAfter
Subject	Name of the subject
Subject public key info	Algorithm identifier and algorithm used to sign certificate
Issuer unique identifier	Contains additional information about the issuer, the exact form of the unique identifier contents is unspecified and is left to the certification authority, it might be, for example, an object identifier, a certificate, a date, or some other form of certification on the validity of the distinguished name; must be version 2 or higher (optional).
Subject unique identifier	Contains additional information about the subject. The exact form of the unique identifier contents is unspecified and is left to the certification authority, it might be, for example, an object identifier, a certificate, a date, or some other form of certification on the validity of the distinguished name; must be version 2 or higher (optional).
Extensions	Optional, An extension field consists of an extension identifier and a criticality flag, When an implementation processing a certificate does not recognize an extension, if the criticality flag is FALSE, it may ignore that extension. If the criticality flag is TRUE, unrecognized extensions shall cause the structure to be considered invalid, i.e. in a certificate, an unrecognized critical extension would cause validation of a signature using that certificate to fail.
Issuer's signature	

Table 5.2: contents of the X-509v3 certificate

5.2.4 Attribute Certificates

User identities are bound to their public key certificates for authentication and identification purposes. Each public key certificate conveys the information necessary to perform certain cryptographic functions. Certified attributes associated with a subject, such as clearances, may be conveyed in a separate structure, defined as an *attribute certificate.*

An attribute certificate is a separate structure from a subject's X.509 public key certificate. A subject may have multiple attribute certificates associated with each of its public key certificates.

For example an attribute certificate could have validity of one day stating that a person has the power to sign a one million dollar contract.

5.3 Public key encryption

Since the invention of public-key cryptography in 1976 by Whitfield Diffie and Martin Hellman [DH76], numerous public-key cryptographic systems have been proposed. All of these systems based their security on the difficulty of solving a mathematical problem [Cer97].

Over the years, many of the proposed public-key cryptographic systems have been broken and many others have been demonstrated to be impractical. Today, only three types of systems are considered both secure and efficient. Examples of such systems and the mathematical problems on which their security is based, are [Cer97]:

1. Integer factorization problem (IFP): RSA and Rabin-Williams.
2. Discrete logarithm problem (DLP): the U.S. government's Digital Signature Algorithm (DSA), the Diffie-Hellman key agreement scheme, the ElGamal encryption and signature schemes, the Schnorr signature scheme, and the Nyberg-Rueppel signature scheme.
3. Elliptic curve discrete logarithm problem (ECDLP): the elliptic curve analog of the DSA (ECDSA), and the elliptic curve analogs of the Diffie-Hellman key agreement scheme, the ElGamal encryption and signature schemes, the Schnorr signature scheme, and the Nyberg-Rueppel signature scheme.

It must be emphasized that none of these problems have been proven to be intractable, i.e., difficult to solve in an efficient manner. Rather, they are believed to be intractable because years of intensive study by leading mathematicians and computer scientists has failed to yield efficient algorithms for solving them [Cer97].

A brief introduction on these methods will be given and the possible future of these systems.

5.3.1 Algorithm based on integer factoring

Factoring is the act of splitting an integer into a set of smaller integers, factors, which, when multiplied together, form the original integer. For example, the factors of 15 are 3 and 5; the factoring problem is to find 3 and 5 when given 15. Prime factorization

requires splitting an integer into factors that are prime numbers; every integer has a unique prime factorization. Multiplying two prime integers together is easy, but as far as we know, factoring the product is much more difficult [RSA96].

Year	*Number of bits*	*MIPS years*
1984	236	.1
1988	352	140
1993	399	825
1994	429	5000
1995	395	250
1996	432	750

The estimate is given in MIPS-Years, where a MIPS-Year is an approximate amount of computation that a machine capable of performing one million arithmetic instructions per second would perform in one year.

There is some variation among published estimates of running time due to the particular definition of a MIPS-Year and to the difficulty of estimating actual processor utilization. (How many arithmetic instructions a modern processor performs in a second when running an actual piece of code depends heavily not only on the clock rate, but also on the processor architecture, the amount and speeds of caches and RAM, and the particular piece of code.)

Table 5.3: Historical data on the integer factorization problem [Cer97]

Factoring is the underlying, presumably hard problem upon which several public-key cryptosystems are based. Factoring an modulus would allow an attacker to figure out the private key; thus, anyone who can factor the modulus can decrypt messages and forge signatures. The security of an factoring algorithm depends on the factoring problem being difficult and the presence of no other types of attack. Unfortunately, it has not been proven that factoring must be difficult, and there remains a possibility that a quick and easy factoring method might be discovered, although factoring researchers consider this possibility remote. Factoring large numbers takes more time than factoring smaller numbers. This is why the size of the modulus determines how secure an actual factoring algorithm is; the larger the modulus, the longer it would take an attacker to factor, and thus the more resistant to attack the an factoring algorithm is [RSA96].

Development in factoring numbers has lead to increases in the number of bits necessary to establish security. Development with the quadratic sieve and the number field sieve method have shown superior factoring qualities. Table 5.3 contains some historical data on the progress of integer factorization.

These results indicate that a 512-bit modulus n provides only marginal security when used in the RSA cryptosystem. For long-term security, 1024-bit or larger moduli should be used [Cer97].

5.3.2 Algorithm based on discrete logarithm problem

The discrete logarithm problem applies to *groups*. A group is an abstract mathematical object consisting of a set G together with an operation $*$ defined on pairs of elements of G. The *order* of a group is the number of elements in G.

If p is a prime number, then the non-zero elements of $Z_n^* = \{0, 1, 2, \ldots, p-1\}$ forms a group of order $p-1$ under the operation of multiplication modulo p. The *order* of a group of elements $g \in G$ is the least positive integer n such that $g^n = 1$. For example, in the group Z_{12}^*, the element $g = 3$ has order 5, since

$$3^1 \equiv 3 \ (mod\, 11),$$
$$3^2 \equiv 9 \ (mod\, 11),$$
$$3^3 \equiv 5 \ (mod\, 11),$$
$$3^4 \equiv 4 \ (mod\, 11), \ and$$
$$3^5 \equiv 1 \ (mod\, 11)$$

The discrete logarithm problem, as first employed by Diffie and Hellman in their key agreement protocol, was defined explicitly as the problem of finding logarithms in the group Z_n^*: given $g \in Z_n^*$ of order n, and given $h \in Z_n^*$, find and integer x, where $0 \leq x \leq n-1$, such that $g^x \equiv h(mod\, p)$, provided that such an integer exists. The integer x is called the *discrete logarithm* of h to the base g.

For example, consider $p = 17$. Then $g = 10$ is an element of order $n = 16$ in Z_{17}^*. If $h = 11$, then the discrete logarithm of h to the base g is 13 because $10^{13} \equiv 11(mod 17)$ [Men].

Factoring of the discrete logarithmic algorithm has precisely the same asymptotic running time as the corresponding algorithm for integer factorization. This can loosely be interpreted as saying that finding logarithms in the case of a k-bit prime modulus p is roughly as difficult as factoring a k-bit composite number n. It is likely safe to say that taking logarithms modulo a 512-bit prime p will remain intractable for the next three or four years. In comparison, a 512-bit RSA modulus will likely be factored within a year or so [Cer97].

These discrete concepts can be extended to arbitrary groups. Let G be a group of order n, and let α be an element of G. The *discrete logarithm problem* for G is the following: given elements α and $\beta \in G$, find and integer $x, 0 \leq x \leq n-1$, such that $\alpha^x = \beta$, provided that such an integer exists.

5.3.3 Algorithm based on elliptic curves

A variety of groups have been proposed for cryptographic use. There are two primary reasons for this [Men]:

1. The operation in some groups may be easier to implement in software or in hardware than the operation in other groups.

2. The discrete logarithm problem in the group may be harder than the discrete logarithm problem in Z_p^*. Consequently, one could use a group G that is smaller than Z_p^*, while maintaining the same level of security.

The above is the case with elliptic curve groups. The result is smaller key sizes, bandwidth savings, and faster implementations. These features are especially attractive for security applications where computational power and integrated circuit space is limited, such as smart cards, PC cards, and wireless devices [Men].

Elliptic curves are mathematical constructions from number theory and algebraic geometry, which in recent years have found numerous applications in cryptography. An elliptic curve can be defined over any field, e.g., real, rational, complex. However, elliptic curves used in cryptography are mainly defined over finite fields. An elliptic curve E over Z_p is defined by an equation of the form

$$y^2 = x^3 + ax + b \tag{5.2}$$

where $a, b \in Z_p$, and $4a^3 + 27b^2 \equiv 0 \,(mod\, p)$ together with a single element denoted O called the "point at infinity," which can be visualized as the point at the top and bottom of every vertical line. The set $E(Z_p)$ consists of all points $(x, y), x \in Z_p, y \in Z_p$, which satisfy the defining equation 5.2, together with O.

Addition of two points on a elliptic curve is defined according to a set of simple rules, e.g., point P plus point Q is equal to point R in figure 5.1. The addition operation in an elliptic curve is the counterpart to modular multiplication in common public-key cryptosystems, and multiple addition is the counterpart to modular exponentiation [RSA96]

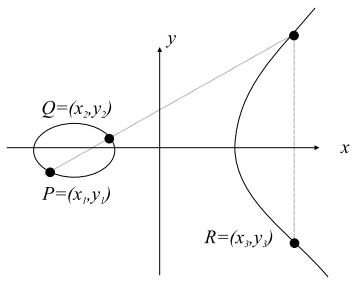


Figure 5.1: Geometric description of the addition of two-distinct elliptic curve points: P+Q=R

Until recently, the best attacks on elliptic curve logarithm problems were the general methods applicable to any group. The methods have a running time of about a constant times the square root of r on average, which is much slower than specialized attacks on certain types of groups. The lack of specialized attacks means that shorter key sizes for elliptic cryptosystems give the same security as larger keys in cryptosystems that are based on discrete logarithm problem. It is possible that algorithm development in this area will change the security of elliptic curve discrete logarithm cryptosystems to be equivalent to that of general discrete logarithm cryptosystems. This is an open research problem [RSA96].

5.4 One-way functions

The definition of a one-way function was given in section 2.7.1 of chapter 2.

A lot depends on the one-way function. If a method can be obtained to develop a plaintext, p_2, that has the same message digest as the original plaintext p_1 then the digital signature can be appended to plaintext p_2. This can be seen in figure 5.2. Because the plaintext p_2 has the same message digest as that of plaintext p_1 the digital signature would be the same for message p_2 as it would be for p_1.

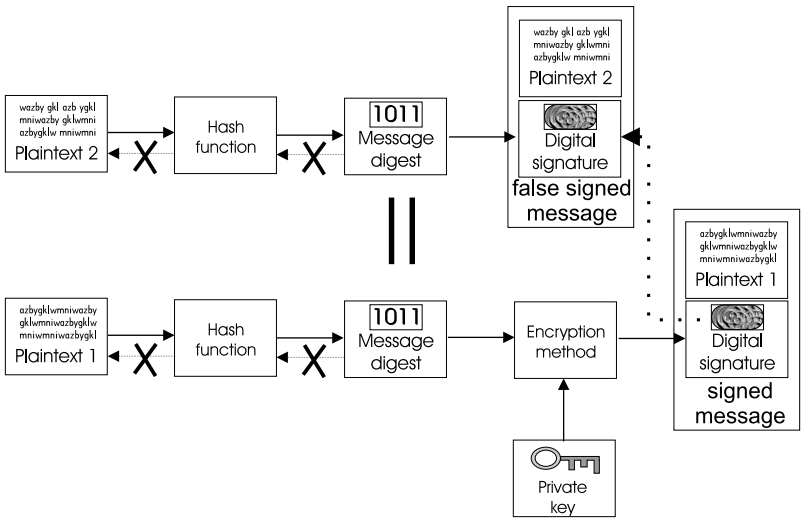


Figure 5.2: Copying the digital signature for another message

Hash functions are designed with a variety of properties in mind and three are commonly singled out in the literature [Rob96]:

1. Given the hash value output by some hash function, it should be infeasible to find an input or pre-image that will produce the given output.

2. When given an input and output pair for some hash function, it should remain infeasible to find a second distinct pre-image that would generate the same output. This is commonly referred to as finding a second pre-image and a hash function for which it is difficult to find either a pre-image or a second pre-image is sometimes called a one-way hash function.

3. It should be infeasible to find two inputs to the hash function that will produce the same output. This is commonly referred to as finding a collision for the hash function.

The term collision-resistant hash function is sometimes used to describe a hash function that possesses all three of the properties described above and it is what most people have in mind when talking about hash functions in general.

Since there are an arbitrary number of possible input strings but only a fixed number of outputs, collisions must exist for a hash function, the objective is to ensure that it is computationally in-feasible to find such examples [Rob96].

Examples of well-known hash functions are MD4, MD5, SHA-1 and RIPEMD-160. MD4 and MD5 were developed by Ron Rivest at MIT for RSA Data Security. They are meant for digital signature applications where a large message has to be "compressed" in a secure manner before being signed with the private key. All three algorithms take a message of arbitrary length and produce a 128-bit message digest [RSA96].

Recent work by Hans Dobbertin has discovered that collisions for MD4 can be found within a few minutes on a typical PC. Even more impressive is the fact that collisions can be constructed, in around an hour, so that the text they represent makes sense [Dob95]. Clearly, MD4 should now be considered broken and should not be used anymore for making a digital signature.

MD5 was developed by Rivest in 1991. It is basically MD4 with "safety-belts" and while it is slightly slower than MD4, it is more secure [RSA96]. At Eurocrypt 96 it was announced that collisions for the compression function of MD5 had been found [Dob96]. Dobbertin demonstrated that collisions for the compression function of MD5 could be found in around 10 hours on a PC [Rob96].

Both algorithms, MD4 and MD5 were submitted to the RIPE consortium[9], which was an EU-sponsored project active between 1988 and 1992 with a goal to propose a portfolio of recommended integrity primitives based on an open call for algorithms. Its independent evaluation of MD4 and MD5 led to the conclusion that these hash functions are less secure than anticipated. As a consequence, the consortium proposed a strengthened version of MD4, which was called RIPEMD. This version was later replaced by RIPEMD-160 designed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. It is intended to be used as a secure replacement for the 128-bit hash functions[10] MD4, MD5, and RIPEMD.

Another alternative to RIPEMD-160 is SHA-1 [Pub93]. This also is a further refinement over the MD5 algorithm. As it stands both RIPEMD-160 and SHA-1 are deemed secure and are the currently recommended hash-functions [IEE98] [Sch98].

At this point all aspects have been dealt with, within the scope of this thesis. What remains is an overview of how the quality aspects of information transfer, as stated in section 2.1 of chapter 2, have securely been taken care of.

[9] RIPE stands for RACE Integrity Primitives Evaluation; the consortium members were C.W.I. (NL) prime contractor, Aarhus University (DK), KPN (NL), K.U. Leuven (B), Philips Crypto B.V. (NL), and Siemens AG (D).

[10] A 128-bit hash result does not offer sufficient protection anymore. A brute force collision search attack on a 128-bit hash result requires 2^{64} or about $2 \cdot 10^{19}$ evaluations of the function. In 1994 Paul van Oorschot and Mike Wiener showed that this brute-force job can be done in less than a month with a $10 million investment. This cost is expected to halve every 18 months.

5.5 Compliance to quality aspects

The model for secure information transfer consisted of the following quality aspects:

- Safe Usage.
- Content integrity;
- Confidentiality;
- Authentication of sender and receiver;
- Non-repudiation of delivery and receipt;

These issues have been dealt with in the following way.

Safe usage	The service should only be available to those users that are authorized. Issues that have covered this aspect concern the key storage in section 5.1.2. The following measures should be taken to insure authorized usage of keys: • *Copying* of a private key should not be possible, without the owning entity finding out; • *Use* of a private key should not be possible, without some private knowledge.
Integrity	The content of data send may not be susceptible to change or at least any change should be identifiable. A digital signature provides a solution to this problem. The signature is message and signer dependent. If the message has been changed then this can be securely assessed with the help of the digital signature. Issues that have covered this aspect concern the public key algorithm, in sections 2.7 and 5.3 and the one-way function, sections 2.7.1 and 5.4. These algorithms should be closely monitored in order to assess the decline in 'security' due to advances in cryptography. German law has drafted a safeguard catalogue, see section **??** [Age97], that describes how individual components should be configured.
Confidentiality	The content of data should be illegible to third parties. A digital envelop, see section 2.9.7 provides a solution to this problem. Issues that have covered this aspect concern the public key algorithm, and the one-way function as discussed above. Other issues that also need to be handled concern the usages of symmetric encryption, see section 2.6 and also very important, abundant amounts of symmetric keys have to be 'produced'. The generation of keys has been dealt with in section 5.1.1.

Authenticity	The origin and the receiver of data should be irrefutably determined. This has been achieved with the help of certificates, this has been discussed in section 3.1 and section 5.2.3. The help of a third party is necessary to securely asses the identity of an entity. Introducing a third party requires many organizational measures to be taken and the introduction of a public key infrastructure, as was explained in chapter 3. The introduction of a trust structure and more specific a third party introduces a great many complex issues. The issue *'trust'* has only briefly been dealt with. Criteria have to be set to quantify trust. These criteria have been translated into legislation, see chapter 4 section 4.2.2.2, section **??** and section 4.4.
Non-repudiation	The sender or receiver of data should not be able to deny having send or received a message. This item was split up into: • non-repudiation of the sender, this issue is handled with the help of the digital signature which irrefutably proofs the sender sent a message; • non-repudiation of the receiver, this issue is handled when the receiver sends a signed reply back to the sender.

5.6 Summary

This chapter has given a more thorough explanation of specific aspects that are very important for public key encryption.

Key generation is very important. This is necessary for the production of public keys but also because of the use of digital envelopes that require a key to be 'produced' for every package. Thus abundant key generation possibilities should be available.

A standard certificate is important will it be possible to communicate with all people on the Internet.

An insight has been given into different public key encryption techniques that are being developed and those that are in use. Due to better understanding of the technique it is possible to 'crack' the methods faster then ever was thought possible. This will require constant attention. Also a new technique, elliptic curve, was introduced which is currently implemented into an IEEE standard. This technique, although young, has a promising future.

Different one-way functions were also shown and the developments that have been made at this end.

At the end of this chapter an overview was given how the quality aspects of information transfer have been securely dealt with. Information from the previous chapters was given to provide a complete view.

Chapter 6

Current implementations

Up to this point all information given, in the previous chapters, was more or less for a theoretical background. The question now remaining is:

How are current implementations adhering to the requirements?

To get a quick look how certificates are implemented into the world today, the easiest way is to start a current e-mail browser[1] They have built in security features that make use of certificates.

6.1 Internet browsers

Figure 6.1 shows the TTP certificates that have been incorporated into the software. With these certificates it is possible to check the certificates received from an entity for correctness. Taking a closer look at the certificate, as shown in figure 6.2.a, reveals information concerning the owner and the issuer, the serial number, the validity of the certificate and the digital fingerprint itself[2]. Furthermore it can be added whether a certificate that has been received from an entity should be accepted for certifying network cites, e-mail or software developers.

With the help of these certificates it is possible to check the correctness of www-pages, e-mail messages and other data received over the Internet. Software that makes use of these certificates and techniques are the following:

- *Secure Socket Layer - SSL*. The SSL protocol is able to negotiate encryption keys as well as authenticate the server before data is exchanged by the higher-level application. The SSL protocol maintains the security and integrity of the transmission channel by using encryption, authentication and message authentication codes.The SSL protocol is application independent, allowing protocols like HTTP, FTP and Telnet to be layered on top of it transparently [RSA96].

[1] This could for instance be Netscape Communicator or Windows Internet Explorer. This chapter will use Netscape Communicator 4.04 as an example.

[2] Note that the issuer and the owner of the certificate are the same. This is because the CA has signed his own certificate. Everything has to start somewhere.

Figure 6.1: CA certificate incorporated in Netscape Communicator

- *Secure Hypertext Transfer Protocol - S-HTTP*. S-HTTP is designed to provide confidentiality, authenticity, integrity, and non-repudiability while supporting multiple key management mechanisms and cryptographic algorithms via option negotiation between the parties involved in each transaction. S-HTTP operates at the application layer. [RSA96].

This method is a great improvement over non-encrypted standard e-mail and web-browsing, because this can provide for secure www-sites and e-mail.

6.1.1 Certificate authorities on the web

A certificate for a web browser can be requested at any of the certification authorities on the web. A sample of CA's currently delivering certificates for web-browsers can be seen in table 6.1 [Net].

VeriSign	http://www.verisign.com
Thawte Consulting	http://www.thawte.com
Societágrave; per i Servizi Bancari - SSB S.p.A.	http://www.ssb.net
Internet Publishing Services	http://www.ips.es
Certisign Certification Digital Ltda	http://www.certisign.com.br
BelSign	http://www.belsign.be

Table 6.1: Certification authorities

These CA's allow for different types of certificates to be created. There is a common differentiation between certificate classes, all providing different security checks.

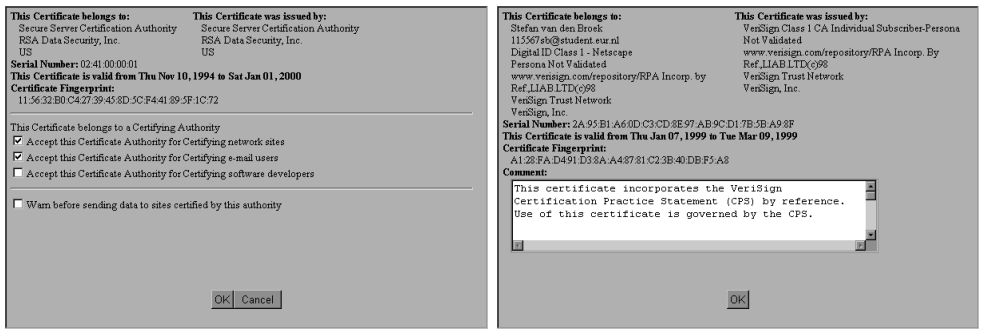


Figure 6.2: Certificate from CA and certificate from entity signed by CA

This can range from checking whether an e-mail address verifying whether a person might have a certain function at an existing company. This al comes at a price, of course.

6.1.1.1 Certificate creation with browser

The certification request procedure, within an browser, is an almost automated process. After selecting the proper certificate class[3] the request, for an free test-certificate with limited value, goes as follows[4]:

1. The user is requested to insert his personal information on the web-site;
2. The public/private key-pair, is automatically generated on the computer of the user;
3. The public key is sent to the CA, while the private key remains on the computer of the user;
4. The CA generates the certificate;
5. The users receives an e-mail with an authentication code to retrieve his certificate;
6. The user requests his certificate at the site with the authentication code from the CA, thus making the certificate available to other users;
7. The certificate is placed on the computer of the user and ready to use.

[3]There exist different certificate classes. Each class provides for a designated level of trust. This depends on the different quality control procedures.

[4]This test was conducted at the VeriSign site. At this site a free test certificate can be obtained that has certain limitations. Class 1 Digital ID costs an Annual fee of US$9.95, Server Digital ID First year: US$349, Commercial Software Publisher (Class 3) Digital ID Annual fee: US$400

This is all done in a secured environment with the help of SSL. For other certificate classes prior contact might be needed. This will allow the CA to check the identity of the end-entity and exchange an identification code with which the end-entity can uniquely identify itself at the certification request.

6.1.2 Shortcomings in the method

Beside the fantastic software that automates almost every part of the process and the added security, there still remain a few shortcomings in this method.

The CA's as described above can be qualified as an off-line TTP[5]. In this method there is no automated process to check for the validity of a certificate. Thus someone might receive a message of which the certificate is compromised. This will not be noticed until the receiver of a message checks in the repository whether the certificate was revoked or not.

Furthermore the sender of a message has no automated means to check whether a message has been received or not.

6.2 A specialized implementation, Netdox

While the method described above provides a great improvement, the shortcomings as described in the precious section do exist. These can be overcome with an in-line TTP[6]. An in-line TTP can verify certificates for there correctness, notarize the time and date a message was sent and provide other features. An implementation that operates as an in-line TTP is *NetDox*[7].

If Alice wants to send a message to Bob, as shown in figure 6.3, she will do the following [8] [9][Net97]:

1. Alice first retrieves the certificate of Bob;
2. Alice then creates a digital envelop for Bob, the inner wrapper as shown in figure 6.3;
3. The software then attaches the digital certificates used in the transaction and waybill information to to the digital envelop. Another digital signature is then made of this larger package using the senders private key. This entire larger package is then re-encrypted with NetDoxs own public key to form a digital envelop. This file is sent over the Internet in a standard format. This double-encrypted file is called a Dox;
4. The Dox is then sent to NetDox;

[5] See section 3.5 chapter 3.

[6] As described in section 3.4 of chapter 3

[7] http://www.netdox.com

[8] Both Alice and Bob of course have to be known by NetDox

[9] Working with NetDox requires the special NetDox program that can be downloaded from the NetDox site.

5. NetDox removes the outer layer of the Dox and checks the authenticity of the message and then archives the digital signature of the original message with a time-stamp indicating when the Dox has been received processed and sent to Bob;

6. NetDox then repackages the inner wrapper with the public key of Bob and sends it to Bob;

7. When Bob receives the message and opens it, a message is sent to NetDox to indicate the package has been opened. This acknowledgment is time-stamped and archived.

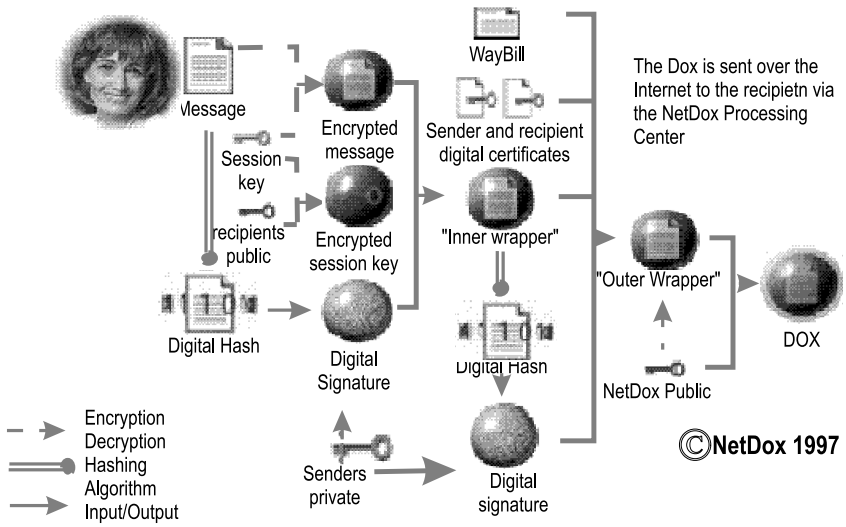


Figure 6.3: Dox going sender to NetDox

NetDox thus always checks the repository to make sure both parties are using valid certificates and thus are making a legal transaction. An automated reply is also sent to NetDox. The added functionality of the process concerns the fact that both parties trust NetDox and NetDox notarizes all actions with a time-stamp. This can be used as evidence in case of a dispute.

6.3 Compliance of implementation with quality aspects

Looking back at the different aspects of information transfer, as stated in section 2.1 of chapter 2, these implementation provide the following:

- *Safe usage*, this issue has not been dealt with completely. In the view of these applications the end-entity is the only person who can use his private key because:

1. He has possession of the private key;
2. He knows the password necessary to use the key;

Legislation however states that it should not be possible to copy the key. This cannot be guaranteed due to the storage of the private key on a disk.

- *Integrity, confidentiality* and *non-repudiation,* this is achieved with help of the digital signature as explained in chapter 2.
- *Authentication,* authentication is provided by the TTP that issues a certificate.

6.3.1 Compliance of current implementation with technical aspects

Current Internet browsers make use of 'weak' encryption, for *confidentiality.* This is due to legislation that has limited the use of strong encryption[10]. *Authentication* occurs in a secure way with a 'good' hash-function and a 'strong' public key, thus creating a 'strong' digital signature which is secure.

NetDox makes use of double encryption. The message is first packed in a digital envelop. Extra information is then added which is then re-packed and encrypted with 'strong' encryption. Should an government body want to have access to a message, then it should have access to the private key of NetDox in order to extract the 'weak' encrypted message[11].

6.3.2 Compliance of current implementations with organizational aspects

To get a good overview of the organizational implementation aspects of a TTP, the best way is to read the Certification Practice Statement. These explain what functions or services are offered and how these should be interpreted. There are a few basic requirements which should be fulfilled, but more important is that what is said actually occurs. This should verified be by an independent auditor and the outcome should be made public.

6.3.3 Compliance of current implementations to legislation

Although not one law exist, there is a broad consensus on how TTP's should manage their organization. These issues have been stated in chapter 4. After reading the CPS one can conclude that much attention has been given to these aspects. It is however difficult to check these issues, given no more then a CPS. This should be left to an external auditor who, as a trusted third party, can verify this.

This brings us to the implementation at the end-entity. Concerning the creation, storage and use of keys, most current implementations store and create these on the computer itself. As stated before this does not comply with legislation[12]. As added security, besides the password protection, some TTP's suggest that the computer on

[10] For instance Netscape offers only 40 bit symmetric encryption. Plans are to increase this value to 128 bit symmetric encryption

[11] This weak encrypted message could be 'cracked' by an government agent.

[12] See section 5.1.2 of chapter 5.

which these keys are stored should be protected against unauthorized access and have virus protection software. In the end however, the TTP states that, it is the responsibility of the end-entity to protect his private key [Ver98].

Recent virus attacks such as Trojan Horses[13], that change a computer into a server when it has an on-line connection with the Internet, make physical attacks perhaps less of a threat. A hacker could attack a computer, contaminated with such a virus, from anywhere in the world without ever having seen or touched the computer.

Currently there are two types of hardware devices available that are more secure than a hard drive for storing a private key and other information[14]. These are known as tokens (typically PCMCIA cards or special floppy disks) and smartcards" [Ver98]. Currently such tokens are not an integral part of current standard implementations[15].

6.4 Summary

This chapter has shown the working of two widely available programs, Netscape Communicator and Windows Internet Explorer and explained the working and shortcomings of these.

A specialized application, NetDox, was then shown that has added functionality and can overcome certain problems associated with the other applications. At this point all aspects that were necessary for secure information transfer have been implemented.

The applications were then tested against the quality aspects of information transfer as stated in section 2.1 of chapter 2. It was shown that they did not completely comply with these requirements. This included the storage of keys, that occurred on a disk.

The three different views, technical, organizational and judicial, were then mapped against the working of the current implementations.

[13] for instance "Back Orifice (BO) is a remote administration system which allows a user to control a computer across a TCP/IP connection using a simple console or GUI application. On a local LAN or across the internet, BO gives its user more control of the remote Windows machine than the person at the keyboard of the remote machine has" [Kas98]

[14] This other information could include the certificate of the CA. This certificate is necessary to check other certificates, web pages and programs.

[15] Netscape communicator 4.04 does have the option of using different cryptographic modules. This could also include a cryptographic hard-ware token.

Chapter 7

Conclusion

The intent of this thesis was to, *provide requirements for secure information transfer across an open digital network, that has such properties that the information sent has legally binding properties.* and furthermore *test current implementations against these requirements.* This had been divided into different aspects: technical, organizational and legal.

This chapter will reflect on the three aspects together that have been dealt with an try to answer whether this solution can be used as evidence in court, because of the combination of these aspects.

7.1 Technical aspects

As stated in section 5.3 of chapter 5, non of the public key encryption methods and one-way functions has proven to be intractable. Rather, they are believed to be. Several legislation[1] have approved to the usage of these methods because as it stand they prove to give sufficient security. The security of these methods will however remain an important issue that has to be evaluated periodically. This is also due to recent developments that have shown ever shorter periods to 'crack' these methods. These developments will probably produce faster 'cracking solutions' in even shorter times, for the following reasons:

- More computing power has become available to mount brute force attacks. This is for the following reasons:
  - Computers are becoming ever faster while the costs are decreasing;
  - Advances in distributed computation make it easier to distribute a problem over several computer[2];

[1] These include Germany and Utah

[2] RSA-laboratories have issued several challenges to crack the DES algorithm. These produced ever faster 'cracking' results that were produced by teams all over the world offering processor time. The latest challenge was won by a team that used nearly 100.000 PC's on the internet and cracked a 56-bit DES algorithm in 22 Hours [RSA99].

- Increased interest in 'cracking' these methods have started specialized methods to be developed[3];

At the other end recent developments have also shown new techniques that prove to be computationally faster but just as strong as RSA encryption[4].

One can conclude that technically much is changing. Methods that are 'safe' today could be 'cracked' within a year. This has to be kept in mind when using this technique. This criticism must not be interpreted as, no secure method exists. Currently several encryption methods do posses a certain amount of security that is 'safe' and has been approved by several government bodies.

7.2 Organizational aspects

Building trust structures is both very complex and costly. First it has to be decided what type of TTP will be developed, an off-line, on-line or in-line TTP. Then what services or functions will be offered. This will then have to be implemented. All these aspects will have to be documented and checked.

At the same time, attention has to be given to legislation. A German TTP must have a license before starting business. A Utah TTP can start business, but without a license the certificates will not have any legal value. In the Netherlands no legislation as of yet exists. A judge may use a message, digitally signed, as evidence, but could also ignore it. The question remains what a Dutch or German judge will do with a digitally signed message that has legal value in Utah.

Another problem concerns standardization. There is an international standard on certificates, the ITU-T X.509. The contents of this certificate standard can however change from CA_1 to CA_2. This is due to the following reasons:

- Non uniformity of contents. CA_1 may insert information concerning e-mail or telephone number while CA_2 does not. There is no unique 'standard' content for a certificate. Certificates can therefore be interpreted differently;
- Interpretation of the contents may vary. While CA_1 may check a name against that of a passport, CA_2 may take a name for granted. Thus the same items listed in a certificate may have different meaning;

Besides the contents of a certificate, although standard, the format in which a certificate is distributed is not standard. Thus a certificate 'understood' by Netscape cannot be imported in Microsoft Internet Explorer[5]. This is awkward to say the least.

One can describe the actions of TTP's as that of jumping into a large swimming pool, without knowing how to swim, and trying to stay afloat. Eventually someone will hopefully stay afloat and others will follow his example, thus setting a standard.

The TTP's adhere to the general public key infrastructure framework as set forth in chapter 3. The general framework is however also under construction. Certain

[3] see section 5.3 of chapter 5, for the decrease in MIPS-years for cracking RSA.

[4] This concerns Elliptic curve encryption

[5] VeriSign for instance offers different export formats for their certificates. This only include those for major formats.

features[6] are still under construction or revision due to problems encountered either in practice or due to legal obligations. This causes the actual implementations to differ and thus obstruct a general method.

7.3 Legislation

Compliance with legislation is an important issue in this thesis. As stated in the introduction, small communities can solve there problems internally, but large communities that have not had any previous contact, must rely on different solutions. Legislation can play a major part in this.

Legislation concerning digital signatures does exist, but it is not uniform. This means a different interpretation can be given to the same digital signature. How these difference will be interpreted remains to be seen. No current jurisprudence exists.

7.4 Conclusion

Looking back at the different aspects one can state the following:

- From a technical point of view, there exists the necessary algorithms[7] to provide for adequate security. The necessary keys can also be generated in a secure way. The implementations at the end-entity however use non-hardware devices to generate and store these keys. Thus the technical implementations at the end-entity have certain flaws;

- From an organizational and judicial point of view, there is broad consensus how a TTP should implement its organization[8]. The actual implementations differ however. Some legislation acknowledges digital signatures while other merely states what should constitute a good solution. Both Utah and German law however state that a legal TTP should have a yearly audit performed to evaluate compliance with this law. The CPS of VeriSign does not mention any audits that have been performed, thus this issue remains uncertain.

Concerning the first issue, the generation and storage of cryptographic material, the German authorities have approved three cryptographic modules. These concern a key generation module, a library function module and a signature token module. All three modules have been produced by Deutsche Telekom AG [fTuPR98]. Approved tokens are thus not widely available and very costly.

Concerning the second issue, the approved CA's, the state Utah has currently recognized only three CA's [Uta99]. VeriSign is not on this list of approved CA's[9]. Receiving a license is very costly. It require extensive procedures to be implemented and should be checked by qualified auditors. Because these procedures are not uniform, in

[6] This includes for instance the Online Certificate Status Protocol (OCSP) and the X.509 certificate that has a current version 3

[7] secret key, public key and hash-functions

[8] These concerns the services such as certificate generation, revocation, production of a repository for certificate, production of a CRL, OCSP et cetera and the technical and personal implementation.

[9] A license only has validity in the state of Utah.

different countries and states, this would require an audit to be performed according to different laws. An almost impossible task.

Providing for a universally recognized digital signatures thus proofs very difficult for the following reasons:

- *Non-uniformity of the technical implementations,* different algorithms are used by different organization. These include both public key- and hash-algorithms. This can make communication between the different applications impossible;
- *Non-uniformity of organizational implementations,* different organizations have implemented PKI-structures that differ in structure. This can lead to different interpretations of the implementations;
- *Non-uniformity of legislation,* different legislation have different interpretations. German law gives a guideline as to what should constitute a 'good' signature, but does not give it a legal status, while Utah law actually gives a digital signature the same status as a hand-written one under certain circumstances.

The above criticism does however not limit the use of the TTP's. Individual TTP's can build good infrastructure that can provide for adequate security within a limited community. These structure can function perfectly without legislation[10]. Within the limits of these structures, and with the help of the the CPS of the different TTP's, different "web of trusts" structures can be built with there own set of rules.

7.5 Future

The future will show at what rate this technology will be incorporated into daily life and how it will be used. A large number of organizations have invested into building a public key infrastructure, but it remains to be seen whether the public will perceive this infrastructure as one to be 'trusted'. A great part of this trust could perhaps be generated with a proper education of the public to understand the technique and learn to use it properly.

The future will perhaps also introduce standard cryptographic tokens formats. This will make these facilities available at low cost to a wide mass of public and enhance the security of digital signatures.

7.6 Afterthought

A great many issues have been dealt with in this thesis but many have not even been mentioned due to the scope of this thesis. Certain issues dealt with have been given limited attention while these issues could have produce enough information to produce an entire thesis. Most issues have however been discussed and the reader should now have a clear understanding of the public key infrastructure and the working of digital signatures.

[10] VeriSign for instance is currently a mayor supplier of digital signatures for a wide range of companies.

7.7 Future research

An important issue that should be dealt with concerns the validity of a digital signatures in time. It is common knowledge that encryption methods have a limited life-span. This will make signatures made in the past susceptible to forgery. Thus signature update procedures will have to be installed or signatures should receive a limited life-cycle. For certain documents limiting the life-cycle is not acceptable. Thus measure should be installed to periodically update the signature and at the same time take such measures that old document formats can still be read.

Appendix A

History encryption

Encryption methods have historically been divided into two categories: substitution ciphers and transposition ciphers.

A.1 Substitution Ciphers

Substitution ciphers preserve the order of the plaintext symbols but disguise them. In a substitution ciphers each letter or group of letters is replaced by another letter or group of letters to disguise it. One of the first methods of encipherment was the *Caesar cipher* devised by Julius Caesar. This method and numerous other methods which have been devised since then were based on the substitution of characters as shown in figure A.1. In this method A becomes N, B becomes O, C becomes P, ... and Z becomes M.

Here k becomes a key to the general method of circular shifted alphabets. The next improvement is to have each of the symbols in the plaintext, say the 26 letters for simplicity, map onto some other letter. The general system is called a *mono-alphabetic substitution,* with the key being the 26 letter string corresponding to the full alphabet.

Mathematically if $m = m_1 m_2 \dots m_n$ is a plaintext message with $m_i \in \Lambda = \{a, b, \dots, z\}$ and ε is a permutation over Λ then the encryption of the message m reads: $E(m) = e(m_1)e(m_2)\dots e(m_n)$

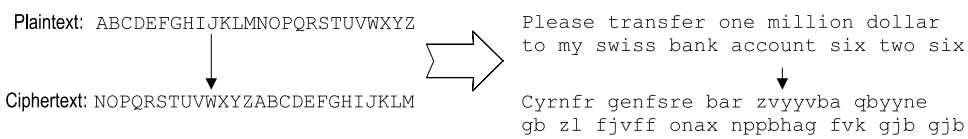


Figure A.1: Caesar cipher

A.1.1 Breaking a substitution cipher

Although the cryptoanalist knows the general system, he does not know which of the $26! = 4 \times 10^{26}$ keys is in use. At 1μsec per solution, a computer would take 10^{13} to try all keys.

Nevertheless, given a surprisingly small amount of ciphertext, the cipher can be broken easily. In English, for example, *e* is the most common letter, followed by *t, o, a, n, i,* etc. The most common two letter combinations, or *diagrams*, are *th, in, er, re* and *an.* The most common three letter combinations, or *trigrams,* are *the, ing, and,* and *ion.*

A cryptanalyst trying to break a monoalphabetic cipher would start out by counting the relative frequencies of all letters in the ciphertext. Then he might tentatively assign the most common one to *e* and the next most common one to *t.* Then he would look at trigrams to find a common one to *e* and the next most common one to *t.*

He would then look at trigrams to find a common one of the form *tXe*, which strongly suggest that *X* is *h.* The search would continue looking for other know trigams like *and.* By making guesses at common letters, diagrams and trigrams and knowing about likely patters of vowels and consonants, the cryptanalyst builds up a tentative plaintext, letter by letter.

A.2 Transposition Ciphers

Transposition ciphers reorder the letters but do not disguise them. The cipher is keyed by a word or phrase not containing any repeated letters. For an example see figure A.2. The purpose of the key is to number the columns, column 1 being under the key letter closest to the start of the alphabet, and so on. The plaintext is written horizontally, in rows. The ciphertext is read out by columns, starting with the column whose key letter is the lowest.

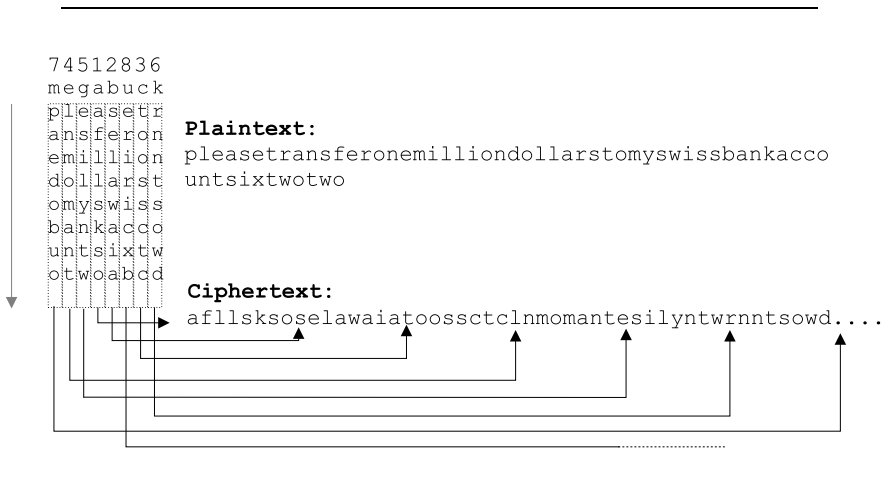


Figure A.2: Transposition encryption

A.2.1 Breaking a transposition ciphers

To break a transposition cipher, the cryptanalyst must first be aware that he is dealing with a transposition cipher. By looking at the frequency of *E, T, A, O, I, N,* etc. it is easy to see if they fit the normal pattern for plaintext. If so the cipher is clearly a transposition cipher, because in such a cipher every letter represents itself.

The next step is to make a guess at the number of columns. In many cases a probable word or phrase may be guessed at from the context of the message. From this the number of columns could be deduced.

The remaining step is to order the columns. When the number of columns, k, is small, each of the $k(k-1)$ column pairs can be examined to see if its diagram frequencies match those for English plaintext. This will be tried and tested for different positions.

A.3 One-Time Pads

Constructing an unbreakable cipher is actually quite easy; the technique has been known for decades. First choose a random bit string as the key. Then convert the plaintext into a bit string, for example by using its $ASCII$ representation. Finally, compute the `exclusive or` of these two strings, bit by bit. The resulting ciphertext cannot be broken, because every possible plaintext is an equally probable candidate. This method is known as the *one-time pad.*

This method however has a number of practical disadvantages:

- The key cannot be memorized, so both sender and receiver must carry a written copy with them;
- The total amount of data that can be transmitted is limited by the amount of key available;
- The method is sensitive to lost or inserted characters or synchronization. If the sender or receiver ever get out of synchronization, then all data from then on will appear garbled.

Traditionally, cryptographers have used simple algorithms and relied on very long keys for their security. Nowadays the reverse is true: the object is to make the encryption algorithm so complex and involuted that even if the cryptanalyst acquires vast amounts of enciphered text of his own choosing, he will not be able to make any sense of it at all.

Transposition and substitutions can be implemented with simple circuits. Figure A.3(a) shows a device, known as a *P-box* (P stands for permutation), used to effect a transposition on a 8-bit input. Substitutions are performed by *S-boxes* , as shown in figure A.3(b). In this example a 3-bit plaintext is entered and a 3-bit ciphertext is output.

The real power of these basic elements becomes apparent when we cascade a whole series of boxes to form a *product cipher,* as shown in figure A.3(c). With the help of these switches encryption can be done at practically the speed of light.

This type of encryption is collectively known as *symmetric* or *secret-key encryption.*

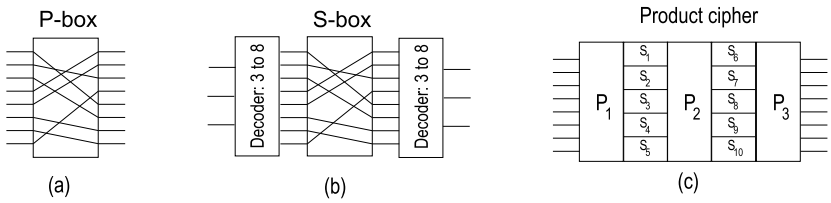


Figure A.3: Basic elements of product ciphers

A.4 Two Fundamental Cryptographic Principles

Although there are numerous different cryptographic systems there are two principles underlying all of them that are important to understand:

1. *redundancy,* all encrypted messages must contain some information not needed to understand the message.

2. *play-back of message*, some measures must be taken to prevent active intruders from playing back old messages.

Redundancy is necessary to prevent active intruders from tricking the receiver into acting on a false message. If no redundancy were added an active intruder could create messages, with some random device, and trick the receiver into believing these messages were genuine.

Also measures should be taken to prevent an active intruder from playing back old messages. If no such measures were taken, an active intruder could keep repeating sending valid messages.

A distinction can be made between two modern encryption methods:

1. Symmetric encryption;

2. Asymmetric encryption or Public Key Encryption (PKE).

Appendix B

The RSA-system

RSA is a public-key cryptosystem for both encryption and authentication. It was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman [RRA77].

The RSA-system is based on the fact that it is easy to compute the product of two prime numbers, but considerably more difficult calculating the two original prime numbers from the product. This is called the trapdoor effect as stated in section 2.7.1.

B.1 Key generation

The generation of key-pair with RSA goes as follows [Tan96]:

1. Generate two large primes, p and q, at random each roughly the same size;
2. Compute $n = p \times q$ and $\theta = (p-1)(q-1)$;
3. Select a random integer $e, 1 < e < \theta$, such that $gcd(e, \theta) = 1$
4. Compute a unique integer $d, 1 < d < \theta$, such that $e \times d = 1 mod \theta$;

The public key now consist of the number-pair (e, n). All other numbers will be kept secret.

B.2 Public key encryption and decryption

If B encrypts a message m for A, which A decrypts, then the process goes as follows:

B should perform the following:

- obtain the public key of A: (e, n);
- represent the message m as an integer m in the interval $[0, n-1]$
- Compute the following:

$$C = m^e (mod\, n) \tag{B.1}$$

The ciphertext is then sent to A. Decryption is almost the same as encryption, except the e is replaced with the d:

$$m = C^d (mod\, n) \tag{B.2}$$

The working of the system is based on the fact that it is almost impossible to calculate d when one only knows (e, n). Calculating d also requires knowledge of p and q. Because one only has n the cryptoanalist has to factor p and q from this.

B.3 Example of RSA encryption

This section will provide an example of RSA encryption.

B.3.0.1 Key generation

Assume the following:

$p = 2357,\ q = 2551.$
Then it follows that: $n = 6012707$ and $\theta = 6007800$;

Choose $e = 3674911$. With $e \times d = 1(mod\ \theta)$ it follows that $d = 422191$
The public key then is: $(6012707, 3674911)$

B.3.0.2 Encrypting and decrypting a message

Encrypting a message, $m = 5234673$, produces the following:
$C = m^e (mod\, n) = 5234673^{3674911}\, mod\, 6012707 = 3650502$

Decrypting the ciphertext, 3650502 produces:
$m = C^d mod\, n = 3650502^{422191}\, mod\, 6012707 = 5234673$

Bibliography

[Act98] Actieplan electronic commerce, March 1998.

[AF98] C. Adams and S. Farrell. Internet x.509 public key infrastructure certificate management protocols. Technical report, PKIX Working Group, may 1998.

[Age97] German Information Security Agency. Bsi manual for digital signatures. http://www.bsi.de, November 1997.

[AR96] A. Abdul-Rahman. The pgp trust model. Internet, augustus 1996.

[Ass96] American Bar Association. *Digital Signature Guidelines.* 750 North Lake Shore Drive, Chicago, IL, august 1996.

[Cer97] Certicom. Remarks on the security of the elliptic curve cryptosystem. Whitepaper, Certicom, 200 Matheson Blvd. W. Mississauga, Ontario L5R 3L7, September 1997.

[Cha96] D.W. Chadwick. *Understanding X.500 - The Directory.* 1996.

[dBC98] Drs. A. de Bos and M.J. Jak RE CISA. *De EDP auditor*, 1998.

[DE94] J. Schiller D. Eastlake, S. Crocker. Randomness recommendations for security. Request for Comments 1750, Network Working Group, December 1994.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[DL96] Kenneth Dam and Herbert Lin. *Cryptography's Role in Securing the Information Society.* National Academy Press, 1996.

[Dob95] H. Dobbertin. Collisions in md4. *CryptoBytes*, 1(3), 1995.

[Dob96] H Dobbertin. Cryptanalysis of md5 compress. Presented at the rump session of Eurocrypt 96, May 1996.

[Dut98] Mr. Drs. A.W. Duthler. *Met recht een TTP!* Kluwer, June 1998.

[DZ83] J.D. Day and H. Zimmerman. The osi reference model. *Proc. of the IEEE*, 71:1334–1340, 1983.

[FIP94] Security requirements for cryptographic modules. FIPS PUB 140-1, National Institute of Standards and Technology (NIST), January 1994.

[FMoET97] Research Federal Ministry of Education, Science and Technology. Informations- und kommunikationsdienste-gesetz - iukdg. *http://www.iid.de*, 1997.

[fTuPR98] Die Regulierungsbehorde fur Telekommunikation und Post (RegTP). Bekanntmachung zur digitalen signatur nach dem signaturgesetz und der signaturverordnung. http://www.regtp.de, September 1998.

[Gif98] David K. Gifford. *Natural Random Number*. MIT/LCS/TM-371, September 1998.

[IEE98] IEEE. Standard specifications for public key cryptography - p1363. draft 3, the Institute of Electrical and Electronics Engineers, Inc - IEEE, May 1998.

[Kam94] Raymond Kammer. Supporting escrowed encryption. http://www.nist.gov/item/testimony/may94/encryp.html, May 1994.

[Kas98] Eugene Kaspersky. Win32.bo, aka back orifice trojan. Metropolitan Network BBS inc. http://www.avp.ch, 1998.

[Knu81] Donald E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*. Addison-Wesley, Reading, MA, 1981.

[Koo97] Bert-Jaap Koops. Crypto regulation in europe, some key trends and issues. *Computer Networks and ISDN Systems*, 29:1823–1831, July 1997.

[Koo98] Bert-Jaap Koops. Crypto law survey. http://cwis.kub.nl/ frw/people/koops/bertjaap.htm, 1998.

[Lex] Lexicon Publication, inc. *Websters Dictionary*, 1991 edition.

[Mat96] Tim Matthews. Suggestions for random number generation in software system,. Bulletin 1, RSA Data Security, January 1996.

[McC97] Candence L. McCuen. Digital/electronic signature state legislative models. Technical report, GTE CyberTrust Solutions Incorporated, Needham, november 12 1997.

[Men] Don B. Johnson Alfred J. Menezes. Elliptic curve dsa (ecdsa): An enhanced dsa. Technical report, Certicom Corp.

[Net] Netscape. Certificate authority services. http://certs.netscape.com.

[Net97] NetDox. Doxit service overview. *http://www.netdox.com*, 1997.

[N.V98] KPMG EDP Auditors N.V. Final report national ttp project. Technical report, Ministry of Economic Affairs, Ministry of Transport, Public Works and Water Management, Amstelveen, 1 March 1998.

[oEC98] Working Group on Electronic Commerce. Draft uniform rules on electronic signatures. Technical Report Thirty-third session, UNITED NATIONS COMMISSION ON INTERNATIONAL TRADE LAW (UNICITRAL), May 1998.

[Poh97] Prof. Dr. Hartmut Pohl. Guidelines for the use of names and keys in a global ttp infrastructure. Technical report, ISIS- Institute for Information Security, may 1997.

[Pub93] Federal Information Processing Standards Publication. Secure hash standard. FIPS 180-1, National Institute of Standards and Technology (NIST), 1993.

[Rob96] Matt Robshaw. Recent results for md2, md4 and md5. *RSA Laboratories' Bulletin*, 4, November 1996.

[RRA77] A. Shamir R.L. Rivest and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Technical report, MIT Laboratory for Computer Science, april 1977.

[RSA96] RSA DATA SECURITY, INC. *Answers to Frequently Asked Questions About Today's Cryptography*, version 3.0 edition, 1996.

[RSA99] RSA. Des challenge iii broken in record 22 hours. http://www.rsa.com/pressbox, January 1999.

[RvdB98] E. Ridderbeekx and J. van den Berg. Internetbeveiliging een beheerperspectief. *Informatie*, (40), april 1998.

[Sch98] Klaus-Dieter Scheurle. Manahmenkatalog nach 16 abs. 6 der verordnung zur digitalen signatur (signaturverordnung - sigv). Technical Report 2.0a, Bundesamtes fr Sicherheit in der Informationstechnik, 1998.

[(SE98] ETSI Technical CommitteeSecurity (SEC). Telecommunications security; electronic signature standardization report. draft TR 101 V0.4.2, European Telecommunications Standards Institute - ETSI, F-06921 Sophia Antipolis Cedex - FRANCE, November 1998.

[SEI98] Seis certificate policy. version 0.93, Secured Electronic Information in Society - SEIS, c/o Post och Telestyrelsen, Box 5398, 102 49 Stockholm Sweden, May 1998.

[Sha63] Claude E. Shannon. *The Mathematical Theory of Communication.* University of Illinois Press, 1963. originally from: Bell System Technical Journal, July and October 1948.

[Sig97] Signaturverordnung - sigv, July 1997.

[Tan96] A.S. Tanenbaum. *Computer Networks.* Prentice Hall, 3 edition, 1996.

[Uta96] Utah digital signature act. http://www.commerce.state.ut.us/web/commerce/digsig/act.htm, 1996.

[Uta99] Utah. Utah licensed certification authorities. http://www.commerce.state.ut.us/, 1999.

[Ver97] Verisign. Verisign certification practice statement. (1.2), May 1997.

[Ver98] VeriSign. Protect your digital id(sm); protect your private key. https://www.verisign.com/repository/PrivateKey_FAQ/#14, 1998.

[X.597] Information technology - open systems interconnection - the directory: Authentication framework itu-t recommendation x.509. Recommendation X.509, ITU-T, June 1997. The identical text is also published as ISO/IEC International Standard 9594-8.

Index