Cryptography 11

11.1 Definition

So much has been said and so much has been gained; thousands of lives have been lost, and empires have fallen because a secret was not kept. Efforts to keep secrets have been made by humans probably since the beginning of humanity itself. Long ago, humans discovered the essence of secrecy. The art of keeping secrets resulted in victories in wars and in growth of mighty empires. Powerful rulers learned to keep secrets and pass information without interception; that was the beginning of cryptography. Although the basic concepts of cryptography predate the Greeks, the present word *cryptography*, used to describe the art of secret communication, comes from the Greek meaning "secret writing." From its rather simple beginnings, cryptography has grown in tandem with technology, and its importance has also similarly grown. Just as in its early days, good cryptographic prowess still wins wars.

As we get dragged more and more into the new information society, the kind of face-to-face and paper-traceable communication that characterized the nondigital communication before the information revolution, the kind of communication that guaranteed personal privacy and security, is increasingly becoming redefined into the new information society where faceless digital communication regimes are guaranteeing neither information and personal security nor personal privacy. Centuries old and trusted global transactions and commercial systems that guaranteed business exchange and payment systems are being eroded and replaced with difficult to trust and easily counterfeitable electronic systems. The technological and communication revolution has further resulted in massive global surveillance of millions of individuals and many times innocent ones by either their governments or private companies; the fight for personal privacy has never been any more fierce, and the integrity and confidentiality of data have become more urgent than ever before. The security and trust of digital transaction systems have become of critical importance as more and more organizations and businesses join

J.M. Kizza, *Guide to Computer Network Security*, Computer Communications
and Networks, DOI 10.1007/978-3-319-55606-2_11

Table 11.1 Modern cryptographic security services

Security services	Cryptographic mechanism to achieve the service
Confidentiality	Symmetric encryption
Authentication	Digital signatures and digital certificates
Integrity	Decryption of digital signature with a public key to obtain the message digest. The message is hashed to create a second digest. If the digests are identical, the message is authentic, and the signer's identity is proven
Nonrepudiation	Digital signatures of a hashed message then encrypting the result with the private key of the sender, thus binding the digital signature to the message being sent
Nonreplay	Encryption, hashing, and digital signature

the e-commerce train. The very future of global commerce is at stake in this new information society unless and until the security of e-commerce can be guaranteed.

Cryptography is being increasingly used to fight off this massive invasion of individual privacy and security, to guarantee data integrity and confidentiality, and to bring trust in global e-commerce. Cryptography has become the main tool for providing the needed digital security in the modern digital communication medium that far exceeds the kind of security that was offered by any medium before it. It guarantees authorization, authentication, integrity, confidentiality, and nonrepudiation in all communications and data exchanges in the new information society. Table 11.1 shows how cryptography guarantees these security services through five basic mechanisms that include symmetric and public key encryption, hashing, digital signatures, and certificates.

A cryptographic system consists of four essential components [1]:

- Plaintext—the original message to be sent
- Cryptographic system (cryptosystem) or a cipher—consisting of mathematical encryption and decryption algorithms
- Ciphertext—the result of applying an encryption algorithm to the original message before it is sent to the recipient
- Key—a string of bits used by the two mathematical algorithms in encrypting and decrypting processes

A cipher or a cryptosystem is a pair of invertible functions, one for encrypting or enciphering and the other for decrypting or deciphering. The word *cipher* has its origin in an Arabic word *sifr,* meaning *empty* or *zero*. The encryption process uses the cryptographic algorithm, known as the encryption algorithm, and a selected key to transform the plaintext data into an encrypted form called ciphertext, usually unintelligible form. The ciphertext can then be transmitted across the communication channels to the intended destination.

A cipher can either be a stream cipher or a block cipher. Stream ciphers rely on a key derivation function to generate a key stream. The key and an algorithm are then applied to each bit, one at a time. Even though stream ciphers are faster and smaller to implement, they have an important security gap. If the same key stream is used, certain types of attacks may cause the information to be revealed. Block ciphers, on the other hand, break a message up into chunks and combine a key with each chunk, for example, 64 or 128 bits of text. Since most modern ciphers are block ciphers, let us look at those in more details.

11.1.1 Block Ciphers

Block ciphers operate on combinations of blocks of plaintext and ciphertext. The block size is usually 64 bits, but operating on blocks of 64 bits (8 bytes) is not always useful and may be vulnerable to simple cryptanalysis attacks. This is so because the same plaintext always produces the same ciphertext. Such block encryption is especially vulnerable to replay attacks. To solve this problem, it is common to apply the ciphertext from the previous encrypted block to the next block in a sequence into a combination resulting into a final ciphertext stream. Also to prevent identical messages encrypted on the same day from producing identical ciphertext, an *initialization vector* derived from a *random number generator* is combined with the text in the first block and the key. This ensures that all subsequent blocks result in ciphertext that doesn't match that of the first encrypting.

Several block cipher combination modes of operation are in use today. The most common ones are described below [2]:

- Electronic Codebook (ECB) mode—this is the simplest block cipher mode of operation in which one block of plaintext always produces the same block of ciphertext. This weakness makes it easy for the cryptanalysts to break the code and easily decrypt that ciphertext block whenever it appears in a message. This vulnerability is greatest at the beginning and end of messages, where well-defined headers and footers contain common information about the sender, receiver, and date.
- Cipher Block Chaining (CBC) mode is a mode of operation for a block cipher that uses what is known as an initialization vector (IV) of a certain length. One of its key characteristics is that it uses a chaining mechanism that causes the decryption of a block of ciphertext to depend on all the preceding ciphertext blocks. As a result, the entire validity of all preceding blocks is contained in the immediately previous ciphertext block. A single bit error in a ciphertext block affects the decryption of all subsequent blocks. Rearrangement of the order of the ciphertext blocks causes decryption to become corrupted. Basically, in cipher block chaining, each plaintext block is XORed (exclusive ORed) with the immediately previous ciphertext block and then encrypted.
- Cipher Feedback (CFB) is similar to the previous CBC in that the following data is combined with previous data so that identical patterns in the plaintext result in different patterns in the ciphertext. However, the difference between CBC and CFB is that in CFB data is encrypted a byte at a time and each byte is encrypted along with the previous 7 bytes of ciphertext.

- Output Feedback (OFB) is a mode similar to the CFB in that it permits encryption of differing block sizes but has the key difference that the output of the encryption block function is the feedback, not the ciphertext. The XOR value of each plaintext block is created independently of both the plaintext and ciphertext. Also like CFB, OFB uses an initialization vector (IV) and changing the IV in the same plaintext block results in different ciphertext streams. It has no chaining dependencies. One problem with it is that the plaintext can be easily altered.

While cryptography is the art of keeping messages secret, *cryptanalysis* is the art of breaking cipher codes and retrieving the plaintext from the ciphertext without knowing the proper key. The process of cryptanalysis involves a cryptanalyst studying the ciphertext for patterns that can lead to the recovery of either the key or the plaintext. Ciphertexts can also be cracked by an intruder through the process of guessing the key.

This is an exhaustive trial-and-error technique which with patience or luck, whichever works first, may lead to the key. Although this seems to be difficult, with today's fast computers, this approach is becoming widely used by hackers than ever before.

The power of cryptography lies in the degree of difficulty in cracking the ciphertext back into plaintext after it has been transmitted through either protected or unprotected channels. The beauty of a strong encryption algorithm is that the ciphertext can be transmitted across naked channels without fear of interception and recovery of the original plaintext. The decryption process also uses a key and a decryption algorithm to recover the plaintext from the ciphertext. The hallmark of a good cryptographic system is that the security of the whole system does not depend on either the encryption or decryption algorithms but rather on the secrecy of the key. This means that the encryption algorithm may be known and used several times and by many people as long as the key is kept a secret. This further means that the best way to crack an encryption is to get hold of the key.

Key-based encryption algorithm can either be symmetric, also commonly known as conventional encryption, or asymmetric, also known as public key encryption. Symmetric algorithms are actually secret key based, where both the encryption and decryption algorithms use this same key for encryption and decryption. Asymmetric or public key algorithms, unlike symmetric ones, use a different key for encryption and decryption, and the decryption key cannot be derived from the encryption key.

11.2 Symmetric Encryption

Symmetric encryption or secret key encryption, as it is usually called, uses a common key and the same cryptographic algorithm to scramble and unscramble the message as shown in Figs. 11.1 and 11.2. The transmitted final ciphertext stream

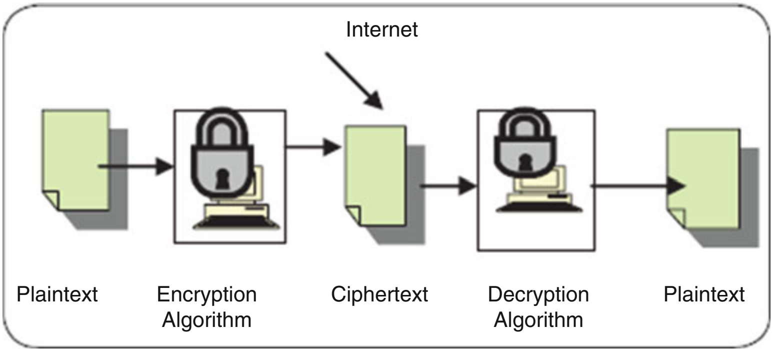


Fig. 11.1 Symmetric encryption

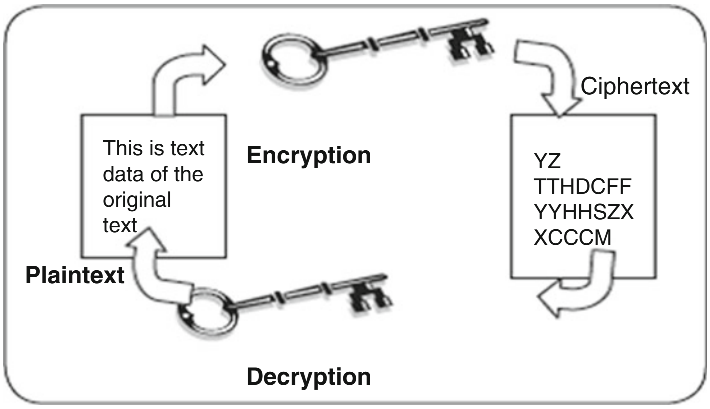


Fig. 11.2 Encryption and decryption with symmetric cryptography

is usually a chained combination of blocks of the plaintext, the secret key, and the ciphertext.

The security of the transmitted data depends on the assumption that eavesdroppers and cryptanalysts with no knowledge of the key are unable to read the message. However, for a symmetric encryption scheme to work, the key must be shared between the sender and the receiver. The sharing is usually done through passing the key from the sender to the receiver. This presents a problem in many different ways, as we will see in Sect. 11.2.2. The question which arises is how to keep the key secure while being transported from the sender to the receiver.

Symmetric algorithms are faster than their counterparts, the public key algorithms.

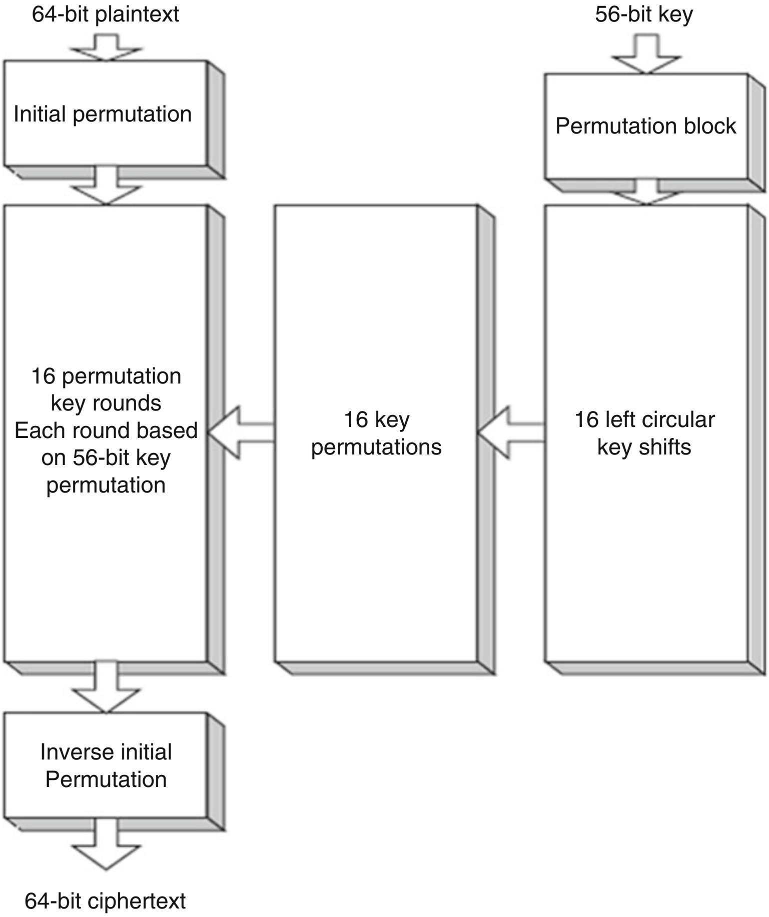

64-bit plaintext
Initial permutation
16 permutation key rounds Each round based on 56-bit key permutation
Inverse initial Permutation
64-bit ciphertext
16 key permutations
56-bit key
Permutation block
16 left circular key shifts

Fig. 11.3 DES algorithm

11.2.1 Symmetric Encryption Algorithms

The most widely used symmetric encryption method in the United States is the block ciphers Triple Data Encryption Standard (3DES). Triple DES developed from the original, and now cracked DES uses a 64-bit key consisting of 56 effective key bits and 8 parity bits. Triple DES encrypts the data in 8-byte chunks, passing it through 16 different iterations consisting of complex shifting, exclusive ORing, substitution, and expansion of the key along with the 64-bit data blocks. Figure 11.3 shows how Triple DES works.

Although 3DES is complicated and complex, and therefore secure, it suffers from several drawbacks including the length of its key fixed at 56 bits plus 8 bits of parity. The limited key length is making it possible for the ever-increasing speed of

Table 11.2 Symmetric key algorithms

Algorithm	Strength	Features (key length)
3DES	Strong	64, 112, 168
AES	Strong	128, 192, 256
IDEA	Strong	64, 128
Blowfish	Weak	32–448
RC4	Weak	
RC5	Strong	32, 64, 128
BEST	Strong	
CAST-128	Strong	32, 128

See more of these algorithms at https://en.wikipedia.org/wiki/Symmetric-key_algorithm

newer computers to render it useless as it possible to compute all possible combinations in the range 0–2^{56} – 1.

Because of this, the National Institute of Standards and Technology (NIST) has presented the Advanced Encryption Standard (AES), which is expected to replace DES. AES is Advanced Encryption Standard whose algorithm was decided to be Rijndael, developed by two Belgian researchers, Joan Daemen and Vincent Rijmen.

Several other symmetric encryption algorithms in use today include International Data Encryption Algorithm (IDEA), Blowfish, Rivest Cipher 4 (RC4), RC5, and CAST-128. See Table 11.2 for symmetric key algorithms.

11.2.2 Problems with Symmetric Encryption

As we pointed out earlier, symmetric encryption, although fast, suffers from several problems in the modern digital communication environment. These are a direct result of the nature of symmetric encryption. Perhaps the biggest problem is that a single key must be shared in pairs of each sender and receiver. In a distributed environment with large numbers of combination pairs involved in many-to-one communication topology, it is difficult for the one recipient to keep so many keys in order to support all communication.

In addition to the key distribution problem above, the size of the communication space presents problems. Because of the massive potential number of individuals who can carry on communication in a many-to-one, one-to-many, and many-to-many topologies supported by the Internet, for example, the secret key cryptography, if strictly used, requires billions of secret key pairs to be created, shared, and stored. This can be a nightmare! Large numbers of potential correspondents in the many-to-one, one-to-many, and many-to-many communication topologies may cause symmetric encryption to fail because of its requirement of prior relationships with the parties to establish the communication protocols like the setting up of and acquisition of the secret key.

Besides the problems discussed above and as a result of them, the following additional problems are also observable:

- The integrity of data can be compromised because the receiver cannot verify that the message has not been altered before receipt.
- It is possible for the sender to repudiate the message because there are no mechanisms for the receiver to make sure that the message has been sent by the claimed sender.
- The method does not give a way to ensure secrecy even if the encryption process is compromised.
- The secret key may not be changed frequently enough to ensure confidentiality.

11.3 Public Key Encryption

Since the symmetric encryption scheme suffered from all those problems we have just discussed above, there was a need for a more modern cryptographic scheme to address these flaws. The answers came from two people: Martin Hellman and Whitfield Diffie, who developed a method that seemed to solve at least the first two problems and probably all four by guaranteeing secure communication without the need for a secret key. Their scheme, consisting of mathematical algorithms, led to what is known as a *public key encryption* (PKE).

Public key encryption, commonly known asymmetric encryption, uses two different keys, a public key known to all and a private key known only to the sender and the receiver. Both the sender and the receiver own a pair of keys, one public and the other a closely guarded private one. To encrypt a message from sender A to receiver B, as shown in Fig. 11.4, both A and B must create their own pairs of keys. Then A and B publicize their public keys—anybody can acquire them. When A has to send a message M to B, A uses B's public key to encrypt M. On receipt of M, B then uses his or her private key to decrypt the message M. As long as only B, the recipient, has access to the private key, then A, the sender, is assured that only B, the recipient, can decrypt the message. This ensures data confidentiality. Data integrity is also ensured because for data to be modified by an attacker, it requires the attacker to have B's, the recipient's, private key. Data confidentiality and integrity in public key encryption are also guaranteed in Fig. 11.4.

As can be seen, ensuring data confidentiality and integrity does not prevent a third party, unknown to both communicating parties, from pretending to be A, the sender. This is possible because anyone can get A's, the sender's public key. This weakness must, therefore, be addressed, and the way to do so is through guaranteeing of sender nonrepudiation and user authentication. This is done as follows: after both A and B have created their own pairs of keys and exchanged the public key pair, A, the sender, then encrypts the message to be sent to B, the recipient, using the sender's private key. Upon receipt of the encrypted message, B, the recipient, then uses A's, the sender's public key to encrypt the message. The return route is also similar. This is illustrated in Fig. 11.5. Authentication of users is ensured because only the sender and recipient have access to their private keys. And

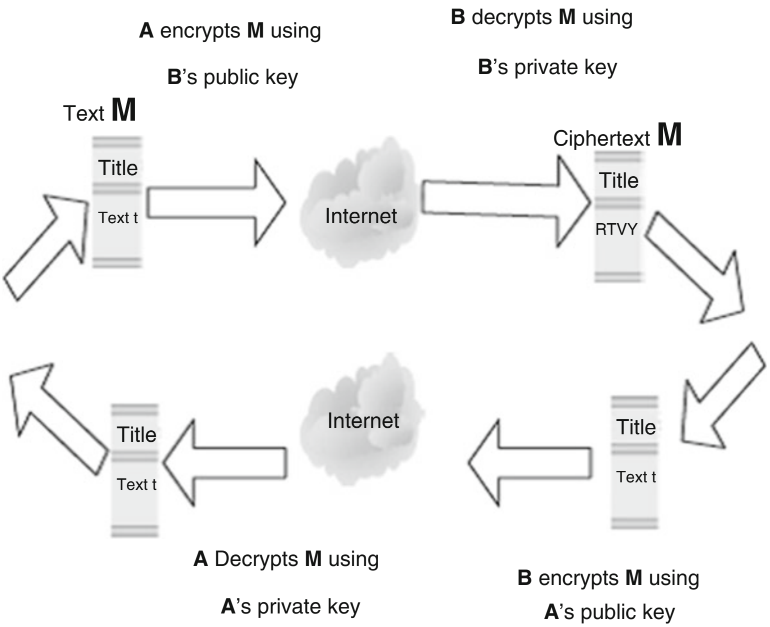


Fig. 11.4 Public key encryption with data integrity and confidentiality

unless their keys have been compromised, both cannot deny or repudiate sending the messages.

To ensure all four aspects of security, that is, data confidentiality and integrity and authentication and nonrepudiation of users, a double encryption is required as illustrated in Fig. 11.6.

The core of public key encryption is that no secret key is passed between two communicating parties. This means that this approach can support all communication topologies including one-to-one, one-to-many, many-to-many, and many-to-one, and along with it, several to thousands of people can communicate with one party without exchange of keys. This makes it suitable for Internet communication and electronic commerce applications. Its other advantage is that it solves the chronic repudiation problem experienced by symmetric encryption. This problem is solved, especially in large groups, by the use of digital signatures and certificates.

The various cryptographic algorithms used in this scheme rely on the degree of computational difficulty encountered as an attempt is made to recover the keys. These algorithms, as we will see in Sect. 11.4, should be labor intensive, and the amount and difficulty involved should, and actually always, increase with the key length. The longer the key, the more difficult and the longer it should take to guess the key, usually the private key.

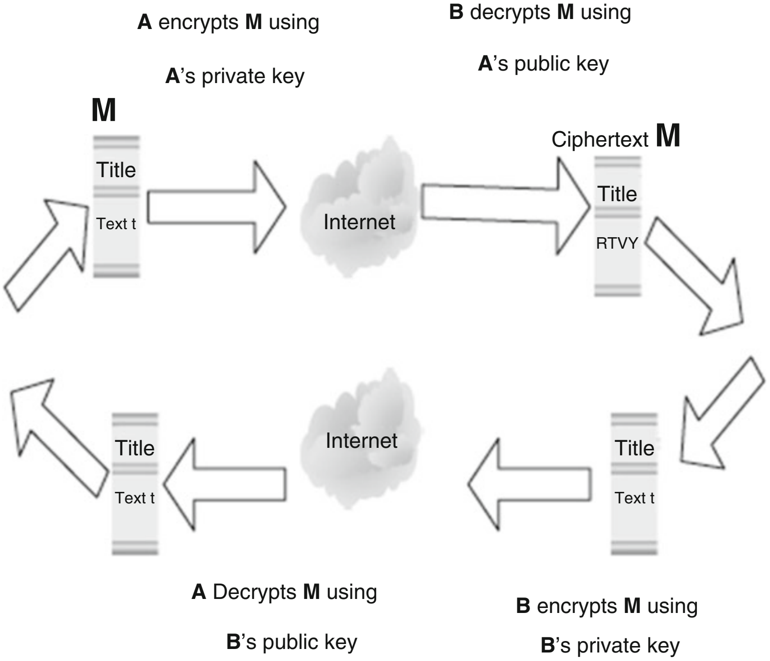


Fig. 11.5 Authentication and nonrepudiation

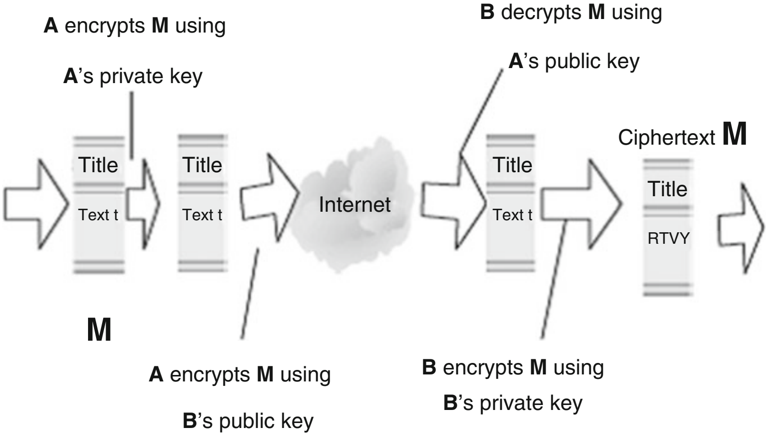


Fig. 11.6 Ensuring data confidentiality and integrity and user authentication and nonrepudiation

11.3.1 Public Key Encryption Algorithms

Various algorithms exist for public key encryption including RSA, DSA, PGP, and El Gamal. Table 11.3 shows the features of such algorithms.

Table 11.3 Public key algorithms

Algorithm	Strength	Features (key length)
RSA	Strong	768, 1024
ElGamal	Strong	768, 1024
DSA	Strong	512–1024
Diffie-Hellman	Strong	768, 1024

See more public key algorithms at https://en.wikipedia.org/wiki/Public-key_cryptography

11.3.2 Problems with Public Key Encryption

Although public key encryption seems to have solved the major chronic encryption problems of key exchange and message repudiation, it still has its own problems. The biggest problem for public key cryptographic scheme is speed. Public key algorithms are extremely slow compared to symmetric algorithms. This is because public key calculations take longer than symmetric key calculations since they involve the use of exponentiation of very large numbers which in turn take longer to compute. For example, the fastest public key cryptographic algorithm such as RSA is still far slower than any typical symmetric algorithm. This makes these algorithms and the public key scheme less desirable for use in cases of long messages.

In addition to speed, public key encryption algorithms have a potential to suffer from the *man-in-the-middle* attack. The man-in-the-middle attack is a well-known attack, especially in the network community where an attacker sniffs packets off a communication channel, modifies them, and inserts them back on to the channel. In case of an encryption channel attack, the intruder convinces one of the correspondents that the intruder is the legitimate communication partner.

11.3.3 Public Key Encryption Services

As it strives to solve the flaws that have plagued other encryption schemes, public key encryption scheme offers the following services:

- Secrecy which makes it extremely difficult for an intruder who is able to intercept the ciphertext to be able to determine its corresponding plaintext. See Fig. 11.4.
- Authenticity which makes it possible for the recipient to validate the source of a message. See Fig. 11.4.
- Integrity which makes it possible to ensure that the message sent cannot be modified in any way during transmission. See Fig. 11.5.
- Nonrepudiation which makes it possible to ensure that the sender of the message cannot later turn around and disown the transmitted message. See Fig. 11.5.

11.4 Enhancing Security: Combining Symmetric and Public Key Encryptions

As we noted in Sect. 11.2.2, symmetric algorithms, although faster than public key algorithms, are beset with a number of problems. Similarly public key encryption also suffers slowness and the potential of the "man-in-the-middle" attacker. To address these concerns and to preserve both efficiency and privacy of the communication channel and increase the performance of the system, a hybrid cryptosystem that uses the best of both and at the same time mitigating the worst in each system is widely used.

11.5 Key Management: Generation, Transportation, and Distribution

One would have thought that the development of advanced technologies would already have solved the chronic problem of exchanging a secret key between two communicating entities. However, one must seriously think that technology is created by humans and humans are part of any technology. But humans also naturally form the weakest links in any technology. They are very unpredictable in what they are likely to do and why they do what they do. Key exchange in cryptographic technologies would not have been a problem, but because of humans, it is.

In a small communication network based on a one-to-one communication topology, the key exchange probably would not be such a problem. However, in modern large networks that support many-to-one, many-to-many, and one-to-many communication topologies, the creation, distribution, and security of millions of keys boil down to a nightmare.

11.5.1 The Key Exchange Problem

In Sect. 11.2.2 we saw that although symmetric encryption is commonly used due to its historical position in the cryptography and its speed, it suffers from a serious problem of how to safely and secretly deliver a secret key from the sender to the recipient. This problem forms the basis for the *key exchange problem*. The *key exchange problem* involves [2] the following:

- Ensuring that keys are exchanged so that the sender and receiver can perform encryption and decryption
- Ensuring that an eavesdropper or outside party cannot break the code
- Ensuring the receiver that a message was encrypted by the sender

The strength of an encryption algorithm lies in its key distribution techniques. Poor key distribution techniques create an ideal environment for a man-in-the-middle attack. The key exchange problem, therefore, highlights the need for strong key distribution techniques. Even though the key exchange problem is more prominent in the symmetric encryption cryptographic methods, and it is basically solved by the public key cryptographic methods, some key exchange problems still remain in public key cryptographic methods. For example, symmetric key encryption requires the two communicating parties to have agreed upon their secret key ahead of time before communicating, and public key encryption suffers from the difficulty of securely obtaining the public key of the recipient. However, both of these problems can be solved using a trusted third party or an intermediary. For symmetric key cryptography, the trusted intermediary is called a *key distribution center* (KDC). For public key cryptography, the trusted and scalable intermediary is called a *certificate authority* (CA). See the side bar in Sect. 9.5.2.2 for a definition of a certificate authority.

Another method relies on users to distribute and track each other's keys and trust in an informal, distributed fashion. This has been popularized as a viable alternative by the PGP software which calls the model the *Web of trust* [2].

11.5.2 Key Distribution Centers (KDCs)

A key distribution center (KDC) is a single, trusted network entity with which all network-communicating elements must establish a shared secret key. It requires all communicating elements to have a shared secret key with which they can communicate with the KDC confidentially. However, this requirement still presents a problem of distributing this shared key. The KDC does not create or generate keys for the communicating elements; it only stores and distributes keys. The creation of keys must be done somewhere else. Diffie-Hellman is the commonly used algorithm to create secret keys, and it provides the way to distribute these keys between the two communicating parties. But since the Diffie-Hellman exchange suffers from the man-in-the-middle attacks, it is best used with a public key encryption algorithm to ensure authentication and integrity. Since all network-communicating elements confidentially share their secret keys with the KDC, it distributes these keys secretly to the corresponding partners in the communication upon request. Any network element that wants to communicate with any other element in the network using symmetric encryption schemes uses the KDC to obtain the shared keys needed for that communication. Figure 11.7 shows the working of the KDC.

Stallings [3] has a very good scenario which describes the working of the KDC, and he describes this working as follows. First both the message sender A and the message receiver B each must have a secret key they each share with the KDC. A initiates the communication process by sending a request to the KDC for a session key and B's secret key. The KDC responds to this request by sending a two-part packet to A. The first part to be sent to A consists of A's request to the KDC, B's secret key, and a session key. The second part, to be sent to B, consists of A's identity and a copy of the session key given to A. Since the packet is to be sent to A,

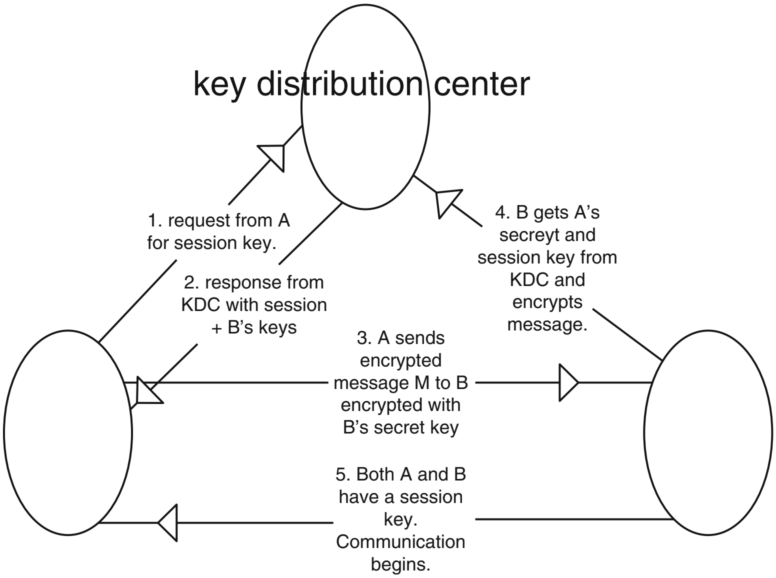

key distribution center
1. request from A for session key.
2. response from KDC with session + B's keys
3. A sends encrypted message M to B encrypted with B's secret key
4. B gets A's secreyt and session key from KDC and encrypts message.
5. Both A and B have a session key. Communication begins.

Fig. 11.7 The working of a KDC

it is encrypted by the secret key the KDC shares with A. When A receives the packet, A then gets out B's secret key and encrypts the message together with B's part of the packet with B's secret key and sends it to B. On receipt, B uses the secret key B shares with the KDC to decrypt the package from A to recover the session key. Now the session key has been distributed to both A and B. After a few housekeeping and authentication handshake, communication can begin.

The KDC has several disadvantages including the following:

- The two network-communicating elements must belong to the same KDC.
- Security becomes a problem because a central authority having access to keys is vulnerable to penetration. Because of the concentration of trust, a single security breach on the KDC would compromise the entire system.
- In large networks that handle all communication topologies, the KDC then becomes a bottleneck since each pair of users needing a key must access a central node at least once. Also the failure of the central authority could disrupt the key distribution system [4].

In large networks with varying communication topologies where network-communicating elements cannot belong to the same KDC, key distribution may become a real problem. Such problems are solved by the Public Key Infrastructure (PKI). We will discuss PKI in Sect. 11.6.

11.5.3 Public Key Management

Because there was a problem with both authenticity and integrity in the distribution of public keys, there was a need to find a solution to this problem. In fact, according to Stallings [3], there were two problems: the distribution of the public keys and the use of public key encryption to distribute the secret key. For the distribution of public keys, there were several solutions including the following:

- Public announcements where any user can broadcast their public keys or send them to selected individuals.
- Public directory which is maintained by a trusted authority. The directory is usually dynamic to accommodate additions and deletions.
- Certificate authority (CA) to distribute certificates to each communicating element. Each communicating element in a network or system communicates securely with the CA to register its public key with the CA. Since public keys are already in public arena, the registration may be done using a variety of techniques including the postal service.

11.5.3.1 Certificate Authority (CA)

The CA then certifies that a public key belongs to a particular entity. The entity may be a person or a server in a network. The certified public key, if one can safely trust the CA that certified the key, can then be used with confidence. Certifying a key by the CA actually binds that key to a particular network-communicating element which validates that element. In a wide area network such as the Internet, CAs are equivalent to the digital world's passport offices because they issue digital certificates and validate the holder's identity and authority. Just as the passport in the real world has embedded information about you, the certificate issued by the CAs has an individual's or an organization's public key along with other identifying information embedded in it and then cryptographically time-stamped, signed, and tamper-proof sealed. It can then be used to verify the integrity of the data within it and to validate this data whenever it is presented. A CA has the following roles [5]:

- It authenticates a communicating element to the other communicating parties that that element is what it says it is. However, one can trust the identity associated with a public key only to the extent that one can trust a CA and its identity verification techniques.
- Once the CA verifies the identity of the entity, the CA creates a *digital certificate* that binds the public key of the element to the identity. The certificate contains the public key and other identifying information about the owner of the public key (e.g., a human name or an IP address). The certificate is digitally signed by the CA.

Since CA verifies the validity of the communicating elements' certificates, it is in charge of enrolling, distributing, and revoking certificates. Because certificates are issued by many different CAs, much of the format of certificates has been defined to ensure validity, manageability, and consistence in the scheme.

To lessen the activities of the CA and therefore improve on the performance of the CA, users who acquire certificates become responsible for managing their own certificates. In doing so, any user who initiates a communication must provide his or her certificate and other identifying information such as a date and random number and send it to the recipient together with a request for the recipient's certificate. Upon receipt of these documents, the recipient sends his or her certificate. Each party then validates each other's certificate, and upon approval by either party, communication begins.

During the validation process, each user may periodically check the CA's lists of certificates which have become invalid before their expiration dates due to key compromise or administrative reasons. Since this may require online access to the CA's central facility, this may sometimes create a bottleneck.

11.5.3.2 Digital Certificates

A digital certificate is a digitally signed message used to attest to the validity of the public key of a communicating element. As we pointed out, digital certificates must adhere to a format. Most digital certificates follow the International Telecommunication Union (ITU-T) X.509 standard. According to RFC 1422, the X.509 digital certificate has the following fields as shown in Table 11.4 and in a sample in Fig. 11.8.

In modern communication, the use of certificates has become common and vital to the security of such communications. For example, in a network environment, in order to encrypt transmissions to your server, the client requires the server's public key. The integrity of that key is vital to the security of the subsequent sessions. If a third party, for example, were to intercept the communication and replace the legitimate key with his or her own public key, that man-in-the-middle could view all traffic or even modify the data in transit. Neither the client nor the server would detect the intrusion.

So to prevent this, the client demands from the server, and the server sends the public key in a certificate signed by a certificate authority. The client checks that digital signature. If the signature is valid, the client knows that the CA has certified that this is the server's authentic certificate, not a certificate forged by a man-in-the-middle. It is important that the CA be a *trusted* third party in order to provide meaningful authentication.

As we will see in Sect. 11.8, when we discuss digital signatures, digital signatures alone cannot authenticate any message and identify a user without a mechanism to authenticate the public key, a role played by the digital certificate. Similarly a digital certificate alone cannot authenticate a message or identify a user without a digital signature. So in order to get a full authentication of a message and identify the user, one needs both the digital signature and digital certificate, both of them working together.

Table 11.4 The ITU-T X.509 digital certificate format [6]

Field	Purpose
Version number	Most certificates use X.509 version 3
Serial number	Unique number set by a CA
Issuer	Name of the CA
Subject issued certificate	Name of a receiver of the certificate
Validity period	Period in which certificate will valid
Public key algorithm information of the subject of the certificate	Algorithm used to sign the certificate with digital signature
Digital signature of the issuing authority	Digital signature of the certificate signed by CA
Public key	Public key of the subject

Several companies now offer digital certificates—that means they are functioning as CAs. Among those are Verisign, American Express, Netscape, US Postal Service, and CyberTrust.

11.5.3.3 Using a Private Certificate Authority

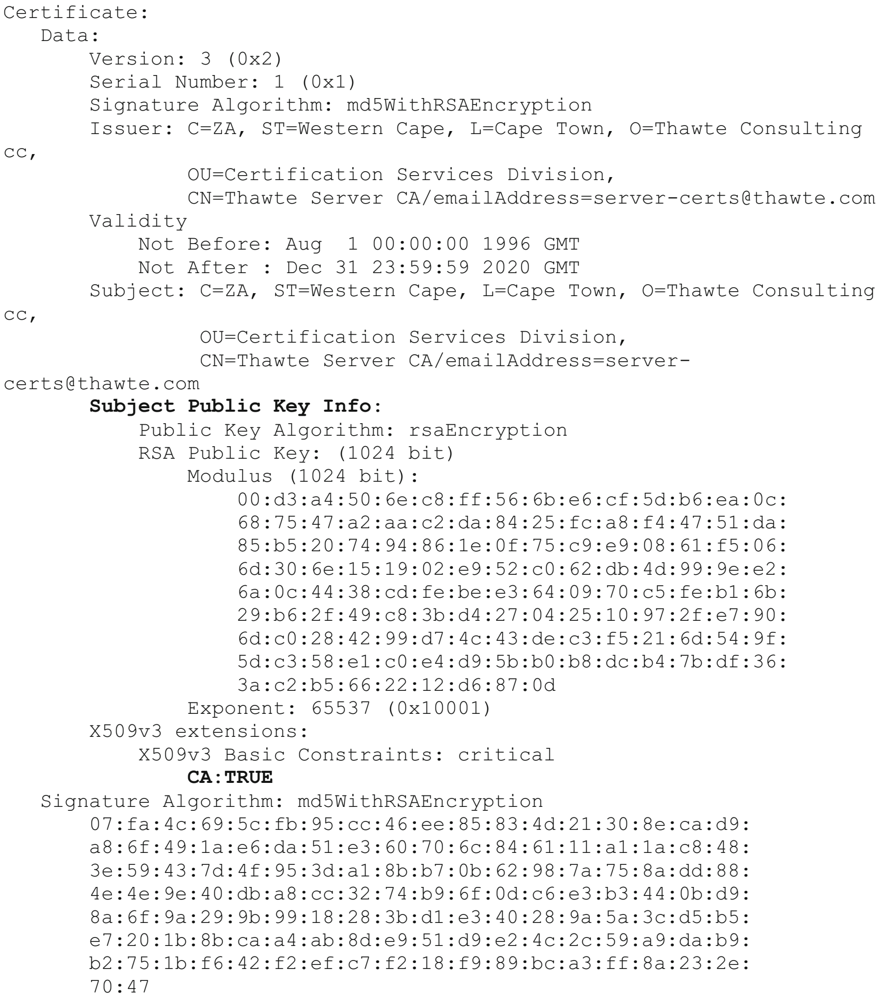
If a business is running its own intranet, it is a security imperative that the security administrator chooses either a public CA or a private CA. It is also possible for the security administrator to create his or her own CA. If one decides to do this, then care must be taken in doing so. One should consider the following steps [7]:

- Consultation with a security expert before building is essential.
- Do all the CA work offline.
- Because it plays a crucial role in the security of the network, it is important that access, both physical and electronic, to the in-house CA must be highly restricted.
- Protect the CA from all types of surveillance.
- Require users to generate key pairs of adequate sizes, preferably 1024 bit.

If the decision is not to use an in-house CA, then it is important to be careful in choosing a good trusted CA.

11.5.4 Key Escrow

Key escrow is a scheme in which a copy of the secret key is entrusted to a third party. This is similar to entrusting a copy of the key to your house or car to a trusted friend. In itself, it is not a bad idea because you can genuinely lose the key or lock it inside the house or car. So in case of the loss of the main key, a copy can always be retrieved from the friend. For private arrangements such as this, the idea of a key escrow is great. However, in a public communication network like the Internet, the idea is not so good. Key escrow began because, as the Internet become more



```
Certificate:
   Data:
       Version: 3 (0x2)
       Serial Number: 1 (0x1)
       Signature Algorithm: md5WithRSAEncryption
       Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting
cc,
               OU=Certification Services Division,
               CN=Thawte Server CA/emailAddress=server-certs@thawte.com
       Validity
           Not Before: Aug  1 00:00:00 1996 GMT
           Not After : Dec 31 23:59:59 2020 GMT
       Subject: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting
cc,
                OU=Certification Services Division,
                CN=Thawte Server CA/emailAddress=server-
certs@thawte.com
       Subject Public Key Info:
           Public Key Algorithm: rsaEncryption
           RSA Public Key: (1024 bit)
               Modulus (1024 bit):
                   00:d3:a4:50:6e:c8:ff:56:6b:e6:cf:5d:b6:ea:0c:
                   68:75:47:a2:aa:c2:da:84:25:fc:a8:f4:47:51:da:
                   85:b5:20:74:94:86:1e:0f:75:c9:e9:08:61:f5:06:
                   6d:30:6e:15:19:02:e9:52:c0:62:db:4d:99:9e:e2:
                   6a:0c:44:38:cd:fe:be:e3:64:09:70:c5:fe:b1:6b:
                   29:b6:2f:49:c8:3b:d4:27:04:25:10:97:2f:e7:90:
                   6d:c0:28:42:99:d7:4c:43:de:c3:f5:21:6d:54:9f:
                   5d:c3:58:e1:c0:e4:d9:5b:b0:b8:dc:b4:7b:df:36:
                   3a:c2:b5:66:22:12:d6:87:0d
               Exponent: 65537 (0x10001)
       X509v3 extensions:
           X509v3 Basic Constraints: critical
               CA:TRUE
   Signature Algorithm: md5WithRSAEncryption
       07:fa:4c:69:5c:fb:95:cc:46:ee:85:83:4d:21:30:8e:ca:d9:
       a8:6f:49:1a:e6:da:51:e3:60:70:6c:84:61:11:a1:1a:c8:48:
       3e:59:43:7d:4f:95:3d:a1:8b:b7:0b:62:98:7a:75:8a:dd:88:
       4e:4e:9e:40:db:a8:cc:32:74:b9:6f:0d:c6:e3:b3:44:0b:d9:
       8a:6f:9a:29:9b:99:18:28:3b:d1:e3:40:28:9a:5a:3c:d5:b5:
       e7:20:1b:8b:ca:a4:ab:8d:e9:51:d9:e2:4c:2c:59:a9:da:b9:
       b2:75:1b:f6:42:f2:ef:c7:f2:18:f9:89:bc:a3:ff:8a:23:2e:
       70:47
```

Fig. 11.8 Sample X.509 certificates (Image source: http://en.wikipedia.org/wiki/X.509)

accessible, wrong characters and criminals joined in with vices such as money laundering, gambling pornography, and drugs. The US government, at least in public, found it necessary to rein in on organized crime on the Internet. The way to do it, as it was seen at that time, was through a key escrow program, and it was hence born.

Since it was first proposed by government, the key escrow program raised a heated debate between those who feel that the program of key escrow is putting individual privacy at risk and those who argue that law enforcement officials must be given the technological ability and sometimes advantage to fight organized crime on the Internet.

The key escrow debate was crystallized by the Clipper chip. The Clipper chip, funded by the U.S. government, was intended to protect private online and telecommunication communications, while at the same time permitting government agents to obtain the keys upon presentation of legal warrant. The government appointed two government agencies to act as the escrow bodies. These agencies were the NIST and the Treasury Department.

The opposition to the Clipper chip was so strong that government was forced to opt for its use to be voluntary.

11.6 Public Key Infrastructure (PKI)

We saw in Sect. 11.5.2 that in large networks with varying communication topologies where network-communicating elements cannot belong to the same KDC, key distribution becomes a real problem. These problems are solved when a Public Key Infrastructure (PKI) is used instead of KDCs to provide trusted and efficient key and certificate management. What then is this PKI? Merike Kaeo, quoting the Internet X.509 Public Key Infrastructure PKIX, defines public key infrastructure (PKI) as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke certificates based on public key cryptography [2]. PKI automate all these activities. PKI works best when there is a large mass of users. Under such circumstances, it creates and distributes digital certificates widely to many users in a trusted manner. It is made up of four major pieces: the certificates that represent the authentication token, the CA that holds the ultimate decision on subject authentication, the registration authority (RA) that accepts and processes certificate signing requests on behalf of end users, and the Lightweight Directory Access Protocol (LDAP) directories that hold publicly available certificate information [8].

11.6.1 Certificates

We defined certificates in Sect. 11.5.3.1 as the cryptographic proof that the public key they contain is indeed the one that corresponds to the identity stamped on the same certificate. The validation of the identity of the public key on the certificate is made by the CA that signs the certificate before it is issued to the user. Let us note here for emphasis that public keys are distributed through digital certificates. The X.509 v3 certificate format, as we noted in Sect. 11.5.3.2, has nine fields. The first seven make up the body of the certificate. Any change in these fields may cause the certificate to become invalid. If a certificate becomes invalid, the CA must revoke it. The CA then keeps and periodically updates the certificate revocation list (CRL). End users are, therefore, required to frequently check on the CRL.

11.6.2 Certificate Authority

CAs are vital in PKI technology to authoritatively associate a public key signature with an alleged identity by signing certificates that support the PKI. Although the CAs play an important role in the PKI technology, they must be kept offline and used only to issue certificates to a select number of smaller certification entities. These entities perform most of the day-to-day certificate creation and signature verification.

Since the CAs are offline and given their role in the PKI technology, there must be adequate security for the system on which they are stored so that their integrity is maintained. In addition, the medium containing the CA's secret key itself should be kept separate from the CA host in a highly secure location. Finally, all procedures that involve the handling of the CA private key should be performed by two or more operators to ensure accountability in the event of a discrepancy.

11.6.3 Registration Authority (RA)

The RAs accept and process certificate signing requests from users. Thus, they create the binding among public keys, certificate holders, and other attributes.

11.6.4 Lightweight Directory Access Protocols (LDAP)

These are repositories that store and make available certificates and certificate revocation lists (CRLs). Developed at the University of Michigan, the LDAP was meant to make the access to X.509 directories easier. Other ways of distributing digital certificates are by FTP and HTTP.

11.6.5 Role of Cryptography in Communication

From our discussion so far, you should by now have come to the conclusion that cryptography is a vital component in modern communication and that public key technology, in particular, is widely used and is becoming more and more acknowledged as one of the best ways to secure many applications in e-commerce, e-mail, and VPNs.

11.7 Hash Function

In the previous sections, we have seen how both symmetric and public key encryptions are used to ensure data confidentiality and integrity and also user authentication and nonrepudiation, especially when the two methods are combined. Another way to provide data integrity and authenticity is to use hash functions.

A hash function is a mathematical function that takes an input message M of a given length and creates a unique fixed-length output code. The code, usually a 128-bit or 160-bit stream, is commonly referred to as a hash or a *message digest*. A one-way hash function, a variant of the hash function, is used to create a signature or fingerprint of the message—just like a human fingerprint. On input of a message, the hash function compresses the bits of a message to a fixed-size hash value in a way that distributes the possible messages evenly among the possible hash values. Using the same hash function on the same message always results in the same message digest. Different messages always hash to different message digests.

A cryptographic hash function does this in a way that makes it extremely difficult to come up with two or more messages that would hash to a particular hash value. It is conjectured that the probability of coming up with two messages hashing on the same message digest is of the order of 2^{64} and that of coming up with any message hashing on a given message digest is of the order of 2^{128} [9].

In ensuring data integrity and authenticity, both the sender and the recipient perform the same hash computation using the same hash function on the message before the message is sent and after it has been received. If the two computations of the same hash function on the same message produce the same value, then the message has not been tampered with during transmission.

There are various standard hash functions of message digest length including the 160-bit (SHA-1 and MD5) and 128-bit streams (RSA, MD2, and MD4). Message digest hash algorithms MD2, MD4, and MD5 are credited to Ron Rivest, while Secure Hash Algorithm (SHA) was developed by the NIST. The most popular of these hash algorithms are SHA and MD5. Table 11.5 shows some more details of these algorithms.

11.8 Digital Signatures

While we use the hash functions to ensure the integrity and authenticity of the message, we need a technique to establish the authenticity and integrity of each message and each user so that we ensure the nonrepudiation of the users. This is achieved by the use of a digital signature.

A digital signature is defined as an encrypted message digest, by the private key of the sender, appended to a document to analogously authenticate it, just like the handwritten signature appended on a written document authenticates it. Just like in the handwritten form, a digital signature is used to confirm the identity of the sender and the integrity of the document. It establishes the nonrepudiation of the sender.

Digital signatures are formed using a combination of public key encryption and one-way secure hash function according to the following steps [10]:

- The sender of the message uses the message digest function to produce a message authentication code (MAC).

Table 11.5 Standard hash algorithms

Algorithm	Digest length (bits)	Features (key length)
SHA-1	160	512
MD5	160	512
HMAC-MD5	Version of MD5	512 (key version of MD5)
HMAC-SHA-1	Version of SHA-1	512 (key version of SHA-1)
PIPEND	160	128

See more hash algorithms at https://en.wikipedia.org/wiki/Secure_Hash_Algorithm

- This MAC is then encrypted using the private key and the public key encryption algorithm. This encrypted MAC is attached to the message as the digital signature.

The message is then sent to the receiver. Upon receipt of the message, the recipient then uses his or her public key to decrypt the digital signature. First, the recipient must verify that the message indeed came from the expected sender. This step verifies the sender's signature. It is done via the following steps [2]:

- The recipient separates the received message into two: the original document and the digital signature.
- Using the sender's public key, the recipient then decrypts the digital signature which results in the original MAC.
- The recipient then uses the original document and inputs it to the hash function to produce a new MAC.
- The new MAC is compared with the MAC from the sender for a match.

If these numbers compare, then the message was received unaltered, the data integrity is assured, and the authenticity of the sender is proven. See Fig. 11.9 for the working of a digital signature verification.

Because digital signatures are derived from the message as a digest which is then encrypted, they cannot be separated from the messages they are derived from and remain valid.

Since digital signatures are used to authenticate the messages and identify the senders of those messages, they can be used in a variety of areas where such double confirmation is needed. Anything that can be digitized can be digitally signed. This means that digital signatures can be used with any kind of message, whether it is encrypted or not, to establish the authenticity of the sender and that the message arrived intact. However, digital signatures cannot be used to provide the confidentiality of the message content.

Among the most common digital signature algorithms in use today are the Digital Signature Standard (DSS) proposed by NIST and based on the ElGamal public key algorithm and RSA. DSS is faster than RSA.

Although digital signatures are popular, they are not the only method of authenticating the validity of the sender and the integrity of the message. Because

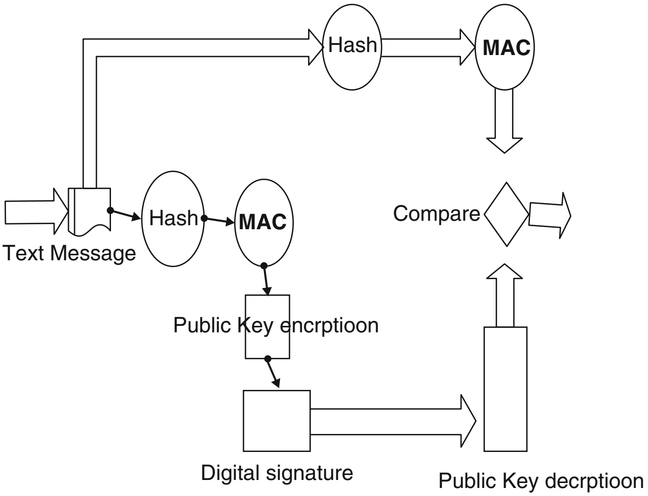


Fig. 11.9 Verifying a digital signature in message authentication

they are very complex, other less complex methods are also in use, especially in the network community. Such methods include the *cyclic redundancy checking* (CRC). In CRC, a digital message is repeatedly divided until a remainder is derived. The remainder, the divisor, along with the message is then transmitted to the recipient. Upon receipt, the recipient would execute the same division process looking for the same remainder. Where the remainder is the same, the recipient is assured that the message has not been tampered with during transmission.

Exercises

1. Discuss the basic components of cryptography.
2. Discuss the weaknesses of symmetric encryption.
3. Discuss the weaknesses of public key encryption.
4. Why is a hybrid cryptosystem preferred over symmetric and public key encryption systems?
5. Why is PKI so vital in modern communications?
6. Discuss the role of digital signatures in modern communication.
7. Some say that with the development of systems such as IPsec, the role the CAs play in modern communication will diminish and eventually cease. Comment on this statement.
8. In a modern communication network, what are the limitations of a tree-structured CA system? Why is it necessary?
9. Discuss the limitations of a KDC system in modern communication.
10. Discuss the future of PKI.

Advanced Exercises

1. Discuss the differences between digital certificates and digital signatures in authentication.
2. Discuss the role and function of a PKI.
3. Describe the sequence of steps a sender of a message takes when sending the message with a digital signature. What steps does the receiver of such a message take to recover the message?
4. Compare and contrast the problems and benefits of KDC and PKI.
5. Describe the message authentication process using:
   (a) Symmetric encryption
   (b) Public key encryption
   (c) Hash function

References

1. Stein LD (1998) Web security: a step-by-step reference guide. Addison-Wesley, Boston
2. Kaeo M (1999) Designing network security. Cisco Press, Indianapolis
3. Stallings W (1999) Cryptography and network security: principles and practice, 2nd edn. Prentice Hall, Upper Saddle River
4. Frame Technology. https://en.wikipedia.org/wiki/Frame_technology_(software_engineering)
5. Key Distribution and Certification. http://mscancer22.tripod.com/securityincomputernetworks/id6.html
6. Panko RR (2004) Corporate computer security. Prentice Hall, Upper Saddle River
7. Certificates and Certificate Authorities. https://en.wikipedia.org/wiki/Certificate_authority
8. Keeping PKI under lock and key. https://www.researchgate.net/publication/295995966_Keeping_PKI_under_lock_and_key
9. Message digests and digital signatures. http://www.diablotin.com/librairie/networking/puis/ch06_05.htm
10. Public key encryption and digital signature: how do they work? http://www.cgi.com/files/white-papers/cgi_whpr_35_pki_e.pdf