NEAR EAST UNIVERSITY

Faculty of Engineerlng

Department of Computer Engineering

Analysis of Cryptography Methods

Graduation Project
COMİlI400

Student: Meryem ŞENOL

Supervisor: Prof.. Dr. Fakhreddln MAMEDOV

Nicosıa-2005

ACKNOWLEDGEMENT

First, I would like to thank to my supervisor Prof. Dr. Fakhreddin MAMEDOV for his available advice, comments on my study. I have successfully made my graduation project, under his guidance and his supervision.

I specially, thank to my fiance, İrfan ATİK, for his encouragement and endless supports to overcome a Iot of difficulties require in my project.

Through the years of education, my family has given me the fortitude to complete my graduate study with their encouragement and supports, special thanks would go them.

ACKNOWLEDGEMENT r

CONTENTS n

ABSTRACT VI

CHAPTER1

SECURITY ATTACKS, SERVICES and MECHANISIMS

1.1 Attacks, Serviees and Mechanisms 1

1.1.1 Services 1

1.1.2 Mechanisms 3

1.1.3 Attacks 5

1.2 Securlty Attaeks 5

1.2.1 Passive Attacks 7

1.2.2 Active Attacks 8

1.3 Seeurity Serviees 9

1.3.1 Confidentiality 9

1.3.2 Authentication 10

1.3.3 Integrity 10

1.3.4 No Repudiation 11

1.3.5 Access Control 11

1.3.6 Availabilty 11

1.4 A Model for Network Seeurlty 11

CHAPTER2

PRIVATE KEY CRYPTOGRAPHYaııd MESSAGE CONFIDENTALLY

2.1 Private Key Encryptioıı 15

2.2 Private Key Cryptography 17

2.3 Cryptanalysis 19

2.4 Prfvate Key Encryption Algorithm 22

2.4.1 Data Encryption Standard 22

2.4.2 Triple Data Encryption Stangard 25

2.4.3 Advanced Encryption Standard 27

2.5 Other Private Key Encryption 28

2.5.1 International Data Encryption Algorithm 28

2.5.2 Blowfish 29

2.5.3 RC5 30

2.5.4 CASJ.. 128 31

2.6 Cipher Bloek Modes of Operatlon 32

2.6.1 Cipher Block Chaining Mode 33

2.6.2 Cipher Feedbaek Mode 36

2.7 Location ofEncryptiônDevices 37

2.8 Key Distribution 39

CHAPTER3

PUBLICKEY CRYTOGRAPHY

3.1 Publie Key Crypfographf Principles 43

3.2 PubUc Key E~cryption Structıfre 44

3.3 Applications for Pubfie Key Cryptosystem 48

3.4 Requirements for Publle Kıey Cryptography 49

3.5 Publie Kıey Cryptography Algorithms 50

3.5.1 The RSA Algorithm 50

3.5.2 Diffie Hellman Key Exchange 55

3.6 Other Publie Key Cryptography Algorithms 56

3.6.1 Digital Signature Standard 56

3.6.2 Elliptic Curve Cryptography 56

3.7 Digital Signaturıes 57

3.8 Kıey Management 58

3.8.1 Digital Certificates 58

3.8.2 Public Key Distribution of Secret Key 59

CONCLUSION 62

REFERENCES 63

ABSTRACT

Cryptography is the science of hiding the contents of messages. Cryptography can be used to provide message secrecy, message integrity, authentication, digital signatures, protection of software and data ete.

This project is developed to the explanation basic cryptography principles. There are two kinds of cryptography methods: private key cryptography and public key cryptography methods. Private Key cryptography methods use the same key to encrypt and decrypt a message, and the public key cryptography methods use one key to encrypt a message anda different key (the private key) to decrypt a message.

And also you can see, in this project, encryption and decryption technique. Encryption encodes a plaintext message by converting it into ciphertext. Decryption is the inverse operation of encryption,

CBAPTER1

SECURITY ATTACKS, SERVICES, AND MECHANISMS

1.1 Attaeks, Serviees, and Meehanlsms

To assess the security needs of an organization effectively and to evaluate and choose various security products and policies, the manager responsible for security needs some systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements. One approach is to consider three aspects of information security:

- Security attaelc Any action that compromises the security of information owned by an organization.

- **Security mechanism:** A mechanism that is. designed to detect, prevent, or recover from a security attack,

- Secarity service: A service thaf enhances the security of the data processing systems and the information transfers ôfaıı ôrğailization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

Serviees

Let us consider these topics briefly, in. reverse order. We can think of information security services as replicating the types of funetions normally associated with phvsıca! documents Much of the activity ôf hmnatıki:rıd,iri· areas as diverse as commerce; foreign policy, military action, aııd persorial ihteractions, depends on the use of doc-uments and On both parties to a transaôtiorı having confidence in the [illegible] of those documents Documents typically have signatures and dates; they may need to be protected from dlsclosure, tampering, or ruction; they may be notarized or witnessed; may be recorded or licensed, and so on.

As information systems become ever more pervasive and essential to the condııct of our affairs, electronic information tak.es on many of the roles traditionally performed by paper documents. Accordingly, the types offunctions traditionally associated with paper documents must be performed on documents that exist in electronic form. Several aspects of electronic documents make the provision of such functions or services challenging:

- Identification
- Authorization
- License and/or certification
- Signature
- Witnessing (notarization)
- Concurrence
- Liability
- Receipts
- Certification of originatioılaııd/orreceipt
- Endorsement
- Access
- Validation
- Time of occurrence
- Authenticity-software files
- Vote
- Ownership
- Registration

Table 1.1 A Partial List of Common Information Integrity Functions [SIMM92b]

It is usually possible to discriminate between an original paper document and a xerograplrlc copy. However, an electronic document is merely a sequence of bits; there is no difference whatsoever between the "original" and any number of copies,

2. An alteration to a paper document may leave some sort of physical evidence of the alteration. For example, an erasure can result in a thin spot or a roughness in the surface. Altering bits in a computer memory or in a signal leaves nö physical trace.

3. Any "proof" process associated with a :physical document typically depends on the physical characteristics of that document (e.g., .the shape of a handwritten signature or an embossed notary seal). Any such proof of authenticity of an electronic document must be based on intemal evidence present in the information itself.

Table 1.1 lists some of the common functions traditionally associated with documents and for which analogous functions for electronic documents and messages are required. We can think of these functions as requirements to be met by a security facility, The list of Table 1.1 is lengthy and is not by itself a useful guide to organizing a security facility. Computer and network security research and development have instead focused on a few general security services that encompass the various functions required of an information security facility.

1.1.2 Meehanisms

There is no single mechanism that will provide ali the services just listed or perform all the functions Iisted in Table 1.1. As this project proceeds, we will see a variety of mechanisms that come into play. However, we can note at this point that there is one particular element that underlies most of the security mechanisms in use: cryptographic techniques. Encryption or encryption-like transformations of information are the most common means ofprovidin.gisecu:rity.

1. Gain unauthorized access to information (i.e. violate secrecy or privacy).
2. Impersonate another user cither to shift responsibility (i.e. liability) or else to use the other' s license for the purpose of:
   - Originating fraudulent information.
   - Modifying legitimate information.
   - Using fraudulent identity to gain unauthorized access.
   - Fraudulently authorizing transactions or endorsing them.
3. Disavow responsibility or Iiability for information the cheater did originate.
4. Claim to have received from some other user information that the cheater created.
5. Claim to have sent to a receiver (at a specified time) information that was not sent (or was sent ata different time).
6. Either disavows receipt of information that was in fact received, or claim a false time of receipt.
7. Enlarge cheater's legitimate license (for access, origination, distribution.etc.).
8. Modify (without authority to do so) the license of others (restrict or enlarge existing licenses.etc).
9. Conceal the presenceofsonıeinfo:rmatiôn (a cove:rt communication) in other information.
10. Insert selfinto a communications link between other users as an active (undeteeted) relay point.
H. Learn who accesses which information (sources, files,etc.) and when the accessesare made cyan if the information itself remains concealed(e.g. a generalization of traffic analysis from communications channels to databases, software, etc.),
12. Impeach an information integrity protocol by revealing information the cheater is supposed to (by the terms of the protocol) keep secret,
13. Pervert the function of software typically by adding covert information.
14. Cause others to violate a protocol by means of introducing incorrect info:rmation.
15. Unilermine confidence in a protocol by causing apparent failures in the system.
16. Prevent communication among other users, in particular surreptitious interference to causeiauthentic communication to be rejected as unauthentic.

Tablel.2 Reasons for Cheating [SIMM92b]

1.1.3 Attaeks

As G.J Simmons perceptively points out, information security is about how to prevent cheating or, failing that, to detect cheating in information-based systems wherein the information itselfhas no meaningful physical existence [SIMM92a].
Table 1.2 lists some of the more obvious examples of cheating, each of which has arisen in a number of real-world cases. These are examples of specific attacks that an organization or an individual (or an organization on behalf of its employees) may need to counter. The nature of the attack that concems an organization varies greatly from one set of circumstances to another. Fortunately, we can approach the problem from a different angle by looking at the generic types of attack that might be encountered.

1.2 Seeurlty Attacks

Attacks on the security of a computer system or network are best characterized by viewing the :functionofthe computer system as providing information. In general,

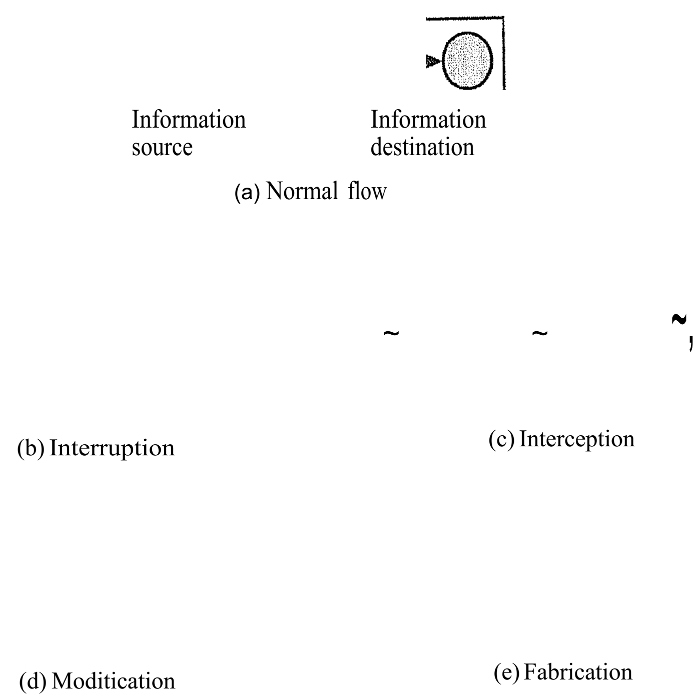


Figure 1.1 Security Threats

There is a flow of information from a source, such as a file or a region of main memory, to a destination, such as another file ora user. This normal flow is depicted in Figure 1.la. The remaining parts of the figure show the following four general categories of attack:

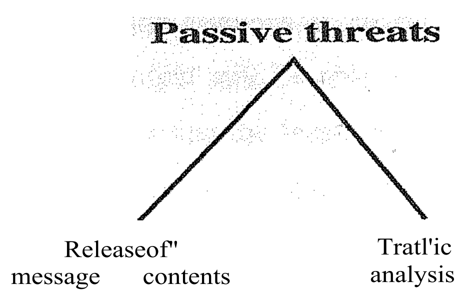
Interruptlom An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability. Examples include destruction ofa piece of hardware, such as a hard disk, the cutting ofa communication line, or the disabling of the file man age management system.

Intereeptien; An unauthorized party gains access to an asset. This is an attack on confidentiality. The unauthorized party could be a person, a program, ora computer. Examples include wiretapping to capture data in a network, and the unauthorized copying of files or programs.

Modification: An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity. Examples include changing values in adata file, altering a program so that it performs differently, and modifying the content of messages being transmitted in a network,

Fabrleanon; An unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity. Examples include the insertion of spurious messages in a network or the addition of records to a file.

useful categorization of these attacks is in terms of passive attacks and active attacks (Figure 1.2).



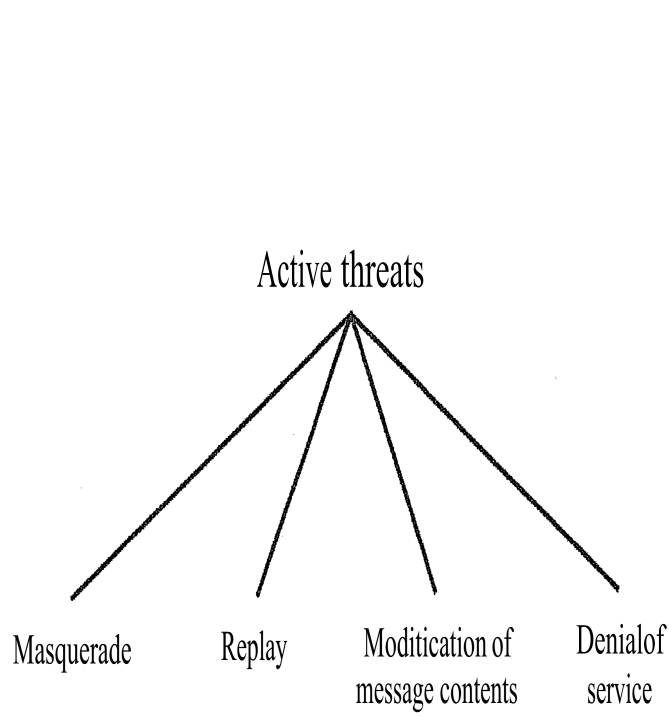


Figure 1.2 Active and Passive Security Threats

1.2.1 Passlve Attacks

Passive attacks are in the hature> of ea:ves dfopping on, or monitoring of, transmission. The goal ofthe opponentisfoobtaillirtformation that is being transmitted, Two types of passive attacks a.re (l)release•of message contents and (2) traffic analysis.

The release of message contents is easily understood. A telephone conversation, an electronie mail message, and a transferred file may contain sensitive or confidential information. We would tike to prevent the opponent from learning the contents of these transmissions.

The second passive attack, traffic analysis, is more subtle; Suppose that we had a way of masking the cB:ıttents ofmessages ôfOther iınomıatfo:ri traffic so that opponents, even if they captured the message; côuld>.ll.ôfiexttactthefufömıation from the message. common technique föi:maskinğ Côıitents is<encfyption. 'If we .had encryption protection in place, an Oppôneiıt :miğhf.still be a.ble to observe the pattern of these messages. The opponent could deter:miıe the focatioll. and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect because they do not involve any alternation of the data. However, it is feasible to prevent the success of these attacks. Thus the emphasis in dealing with passive attacks is on prevention rather than detection.

1.2.2 Aetlve Attaeks

The second major category of attack is active attacks. These attacks involve some modification of the <lata stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

A masquerade takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entitythathas those privileges.

Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect. For example, a message meaning "Allow John Smith to read confidential file accounts" is modified to mean "Allow Fred Brown to read confidential file accounts,"

The denial of service prevents or inhibits the normal use or management of communications facilities, Tfüs attack may have a specific target; for example, an entity may suppress all messages directed to a pajifou.lar destination (e.g., the security audit service). Another form of service deniaLisjhedisruption of an entire network, either by disabling the network or by overloadingit withnıessages so as to degrade performance.

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are dif:ficultto detect, measures are available to prevent their success.

On the other hand, it is quite difficult to prevent active attacks absolutely, because to do so would require complete protection of all communications facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them. Because the detection has a deterrent effect, it may also contribute to prevention.

1.3 Seeurity Serviees

üne useful classification of security services is the following:

- Confidentiality
- Authentication
- Integrity
- No Repüdiation
- Access Control
- Availability

1.3.1 Confidentiality

Confidentiality is the protection of transmitted data from passive attacks. With respect to the release of message contents, several levels of protection can be identified. The broadest service protects all usefdata traı:ısmittedbetween two users over a period of time. For example, ifa virtual circuit is set up between two systems, this broad protection would prevent the release of any user data transmitted over the virtual circuit, Narrower forms of this service can also, be defined, including the protection ofa single message or everispecific fields within a message. These refinements are less useful than the broad approacffandmay even be more cornpfox aı:ıd expensive' to implement,

The other aspect of coıfideritiality is the .protection of traffic flow from analysis. This requires that an attacker .iıôt be abl~ tô Öbserve the source and destination, frequency, length, or other characteristics ôf the traffic ona comrnımications facility.

1.3.2 Authentication

The authentication service is concemed with assuring that a communication is authentic. in the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from. In the case of an ongoing interaction, such as the connection ofa terminal to a host, two aspects are involved.

First, at the time of connection initiation, the service assures that the two entities are authentic (that is, that.each is the entity that it claims to be).

Second, the service must assure that the connection is not interfered with in such a way that a third party can masquerade as one of the two legitimate parties for the purposes of unaııthorized transmission or recepti on.

1.3.3 Integrity

As with confidentiality, integrity cana.pply.to a stream of messages, a single message, or selected fields within a ttıessage. Again, the most useful and straight forward approach is total stream protection.

A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent, with no duplication, insertion, .modification, reordering, or replays. The destruction of data is also covered under this service. Thus, the connection-oriented integrity service addresses both message stream modification and denial of service. On the other hand, a connectionless integrity service, one that deals with individual messages only without regard to any larger context, generally provides protection agai:risf.rrı~ssı:ığet1 1ociification only.

We can make a distinctiollbefyveefrtheService with and without recovery. Because the integrity service relates to active attacks, we are concemed with detection rather prevention. If a violation of integrity is detected, then the service may simply report this violation, and some other portion of software or human intervention is required to recover from the violation. Alternatively, there are mechanisms available to recover from the loss of integrity of data, as we will review subsequently. The incorporation of automated recovery mechanisms is, in general, the more attractive altemative,

1.3.4 No Repudiation

No repudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the message was in fact sent by the alleged sender. Similarly, when a message is received, the sender can prove that the message was in fact received by the alleged receiver.

l.3.5 Aeeess Centrol

in the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this .control, each entity trying to gain access m.ust firstbe İdentified, or authenticated, so that access rights cart be tailored fo the iii.dividual.

1.3.6 Availability

A variety of attacks can result in the loss of or reduction in availability. Some of these attacks are amenable to automated countermeasures, such as authentication and encryption, whereas others require some sort of physical action to prevent or recover from loss of availability of elements ofa distributed system.

1.4 A ModelfOr NetwôrRSecürity

A model for muchof whafwe wi.llbeüiscussinğis captured, in very general terms, in Figure 1.3. A message is to be tra:nsferredfrôm ône party to another across some sort of internet.

The two parti es, who are the *princtpai«* in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the İnternet from source to destination and by the cooperative use of communication protocols (e.g., *TCPIIP)* by the two principals.

Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

- A security-related transformation on the information to be sent. Examples include the encryption ofthe message, which scrambles the message so that,

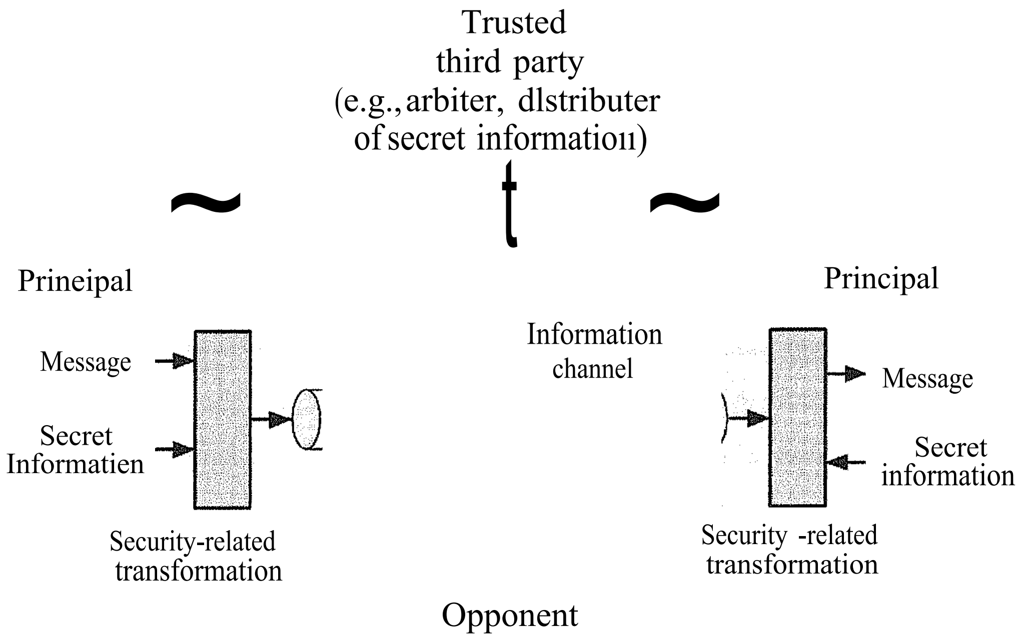


Figure 1.3 A Model for Network Security

it is unreadable by the opponent, and the addition ofı:ı code based on the contents of the message, Which can.be used to verify theidentityof fhe sender, Some secret information shared |>y tlıe two.prirıcipaJ.suıcl,jtis hoped, .u:rıknown to the opponent. An example is an encryption kefusedincörıjurıction withthetransformation to scramble the message before transmission and tıtıscramble it ön reception.

A trusted third party may be needed tö achieve secure transmission. For example, a party may be responsible for distributing the secret information to the two principals while keeping it from any opponents. Or a third party may be needed to arbitrate disputes between the two principal s concerning the authenticity of a message transmission. This general model shows that there are four basic tasks in designing a particular security service:

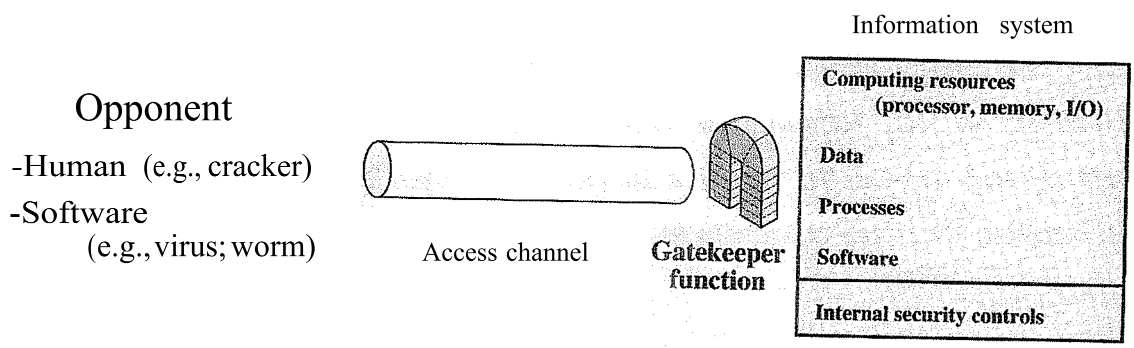
1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose,

2. Generate the secret information to be use d with the algorithm .

3. Develop methods for the distribution and sharing ofthe secret information.

4. Specify a protocol-to be used by the two principals that makes use of the security algorithm and the secret'lnformation to achieve a particular security service.

The types of security mechanisms and services that fit into the model shown in Figure 1.3. However, there are other security.related sitüations of interest that do not neatly fit this model but that are conside:redirithis boôk: A general model of these other situations is illustrated by Figure 1.4, which teflects a concern for protecting an information system from unwanted access Most readers are familiar with the concems caused by the existence ofhackers, who



Ffgure 1.4 Network Access Security Model

Attempt to penetrate systems that can be accessed over a network. The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system. Or, the intruder can be a disgruntled employee who wishes to do damage, or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers),

Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and oompilers, Two kinds of threats can be presented by programs:

Infermation aeeess threats intercept or modify <lata on behalf of users who should not have access to that data,

Service threats exploit service flavvşiincol11pµtersjo Inhibit use by Iegitimate users. Viruses and wormssare two examples ôf şoftware attacks. Such attacks can be introduced into a system by means of a. diskette that contains the unwanted logic concealed in otherwise useful software. ·They caıtalso be inserted into a system across a network; this latter mechanisrnis ofmore concem.in-network security.

The security mechanisms needed to cope with unwanted access fall into two broad categories. The first category might be termed a gatekeeper function. It includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to deny and reject worms, viruses, and other similar attacks.. ünce access is gained, by either .an unwanted user or unwanted software, the second line of defense consists of a variety of internal controls that monitor activity .and arıalyz;est()redinformationina.n.attempt to detect the presence of unwanted intruders.

CHAPTER2

PRIVATE KEY CRYTOGRAPHY AND MESSAGE CONFIDENTALLY

2.1 Prrvate Key Eneryption

Private Key encryption, also referred to as conventional encryption, or symmetric encryption secret-key, or single-key encryption, was the only type of encryption in use prior to the development of public-key encryption in the late 1970s.

This form of encryption was used by Julius Caesar, the Navaho Indians, German U-Boat commanders to present day.military, government andprivate sector applications, it equires all parties that are communicating.tc share a common key. It remains by far the most widely used of the two types ofen.cryption.

This chapter begins with a look at a general model for the private key encryption process; this will enable us to understand the context within which the algorithms are used. Then we look at three .importanr encryption algorithms; Data Encryption Standard (DES), triple DES, and Advance Encryption-Standard (AES) and also we look atanother encryption algorithms; International Data Encryption (IDEA), Blowfish; RC5, CAST-128.

A private key encryption.scheme has five ingredients (Fiğilie 2.1):

- Plalntextı This is the ôriğinal rnessage .or data füatis fed into the algorithm as input.

- Eneryption Algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.

- Seeret Key: The secret key is also input to the algorithms. The exact substitutions and transformations performed by the algorithm depend on the key,

- Cıphertexe This is the scrambled message produced as output, it depends on the plaintext and the secret key. Fora given message, two different keys will produce two different cipher texts.

- Decryption Algorithm: This is essentially the encryption algorithm run in reverse. it takes the cipher text and the same secret key and produces the original plaintext.

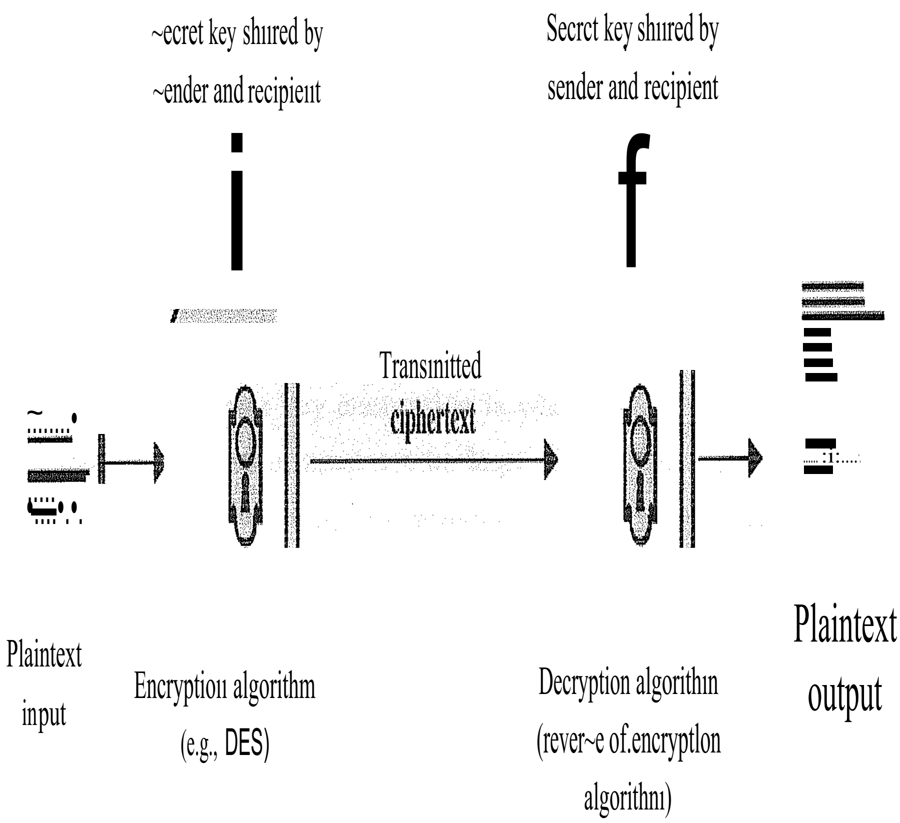


Figure 2.1 SimplifiedModel Of Private Key Encryption

There are two requirements for secure use of private key encryption:

1. We need a strong encryption algorithm. At a minimum, we would like the algorithm to be such that an opponent who knows the algorithm and has access to one or more cipher texts would be unable to decipher the cipher text or figure out the key. This requirement is usually stated in a stronger form:

2. The opponent should be unable to decrypt cipher text or discover the key even if he or she is in possession of a number of cipher texts together with the plaintext that produced each cipher text.

3. Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure. If someone can discover the key and knows the algorithm, all communication using this key is readable.

it is importantto note thatthe security of private key encryption depends on the secrecy of the key, not the sectecyöfthe.algorithm.That is, it is assumed that it is impractical to decrypt a message on theibasis ôfthe ciphertext *plus* knowledge of the encryption/decryption algorithm. In Othefwörds, we<do not need to keep the algorithm secret; we need to keep only the key secret.

This feature of private key encryption is what makes it feasible for widespread use. The fact that the algorithm need not be kept secret means that manufacturers can and have developed low-cost chip implementations of <lata encryption algorithms. These chips are widely available and incorporated into a number ofproducts. With the use ofprivate key encryption, the principal security problem is rnaintaining the secrecy ofthe key.

2.2 Private Key Cryptography

Private key cryptography is often used to encrypt data on hard drives. The pP.r~onencrypting the data holds the key privately and there is no problem with key distribution.

Private key cryptography is also used for communication devices like bridges that encrypt all data that cross the link. A network administrator programs two devices with the same key, and then personally transports them to their physical locations.
If secret-key cryptography is used to send secret messages between two parties, both the sender and receiver must have a copy of the secret key. However, the key may be compromised during transit. If you know the party you are exchanging messages with, you can give them the key in advance. However, if you need to send an encrypted message to someone you have never met, you'll need to figure out a way to exchange keys in a secure way. üne method is to send it via another secure channel or even via overnight express, but this may be risky in some cases.

Cryptographic systems are generically classified along three independent dimensions:

1. The type of operations used for transforming plaintext to ciphertext. All encryption algorithms are basedon nv~;~e~~raliprinciples: substitution, in which each element in the plaintext (bit, letter, .groμp pfbits .pr.letters) is mapped into another element, and transposition, in which elemen.tsin the plaintext are rearranged. The fundamental requirement is that no information be Iost (that is, that all operations be reversible). Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions.

2. The number of keys used. ΓΓ both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key or conventional encryption. If the senderand receiver each use a different key, the system is referred to as asymmetric, two-key, or public-key encryption.

3. The way in which the plaihtext is processe&Ablockcipher processes the input one block of elements at a time, prq<i.μcing an. output block for each input block, A stream cipher processes the inpufelemen.tscôııtinuôusly, producing output one element at a time, as it goes along.

2.3 Cryptanalysis

The process of attempting to discover the plaintext, or key is known as cryptanalysis, The strategy used by the cryptanalyst depends on the nature of the encryption scheme and the information available to the cryptanalyst.

Type of Attaek	Known to Cryptanalyst
Ciphertext only	. Encryption algorithm . Ciphertext to be decoded
Known plain text	. Encryption algorithm . Ciphertext to be decoded .. O.ne. or more plaintext-ciphertext pairs formed with the secretkey
Chosen plaintext	. Ertcryption 'algorithm . Ciphertext to be decoded . Plaintext message chosen by crypt analyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	. Encryption algorithm . Ciphertext to be decoded . Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plain text generated with the secret key
Chosen text	. Encryption algorithm . Ciphertext to be decoded . Plaintext message chosen by cryptiüialyst, together with its corresponding ciphertextgeriefütedwith the secret key . Purported••••cip~e~ext<<ch?sehby··· cryptanalyst, togethed lwith its corresponding•.decrypted plain text generated with ~he secret key

Table 2.1 Types of Atta.cks onEnerypted Messages

Table 2.1 summarizes the various types of cryptanalytic attacks, based on the amount of information known to the cryptanalyst, The most difficult problem is presented when all that is available is the ciphertext only. In some cases, not even the encryption algorithm is known, but in general we can assume that the opponent does know the algorithm used for encryption. One possible attack under these circumstances is the brute-force approach trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it. To use this approach, the opponent must have some general idea of the type of plain text that is concealed, such as English or French text, an MS-DOS EXE file, a Java source listing, an accounting file, and so on.

The ciphertext-only attack is the easiest to defend against because the opponent has the least amount of Informaıion to work with. In many cases, however, the analyst has more informations.Fhe analyst·.: may be ', able to capture one or more plaintext messages as well as their encryptiöns; Or the analyst may know that certain plain text pattems will appear in a message. For ex:aınple,?a file that is encoded in the Postscript format always begins with the same pattem, or there rn.ay be a standardized header or banner to an electronic founds transfer message, and so on. AH these are examples of known plaintext With this knowledge, the analystrn.ay be able to deduce the key on the basis ofthe way in whichthe knovvıı.plaintextis•transformed.

Closely related to the known-plaintext attack is what might be referred to as a probable-word attack, If the opponent is working with the encryption of some general prose message, he or she may have little knowledge ofwhat is Inthe message.
However, if the opponent is after some very specific information, then parts of the message may be known. For example, ifan entire accounting file is being transmitted, the opponent may know the placefüentofcertain.key.vııords.in.theheader ofthe file. As another example, the source cöde fori a p:rograni>developedby a corporation might include a copyright statement in.sofüestiındardized position,
If the analyst is able somehow to get' theesource system to insert into the system a message chosen by.the analyst, then a chosen..plaintext attack is possible. in general, if the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure ofthe key.

Table 2.1 lists two other types of attack: chosen ciphertext and chosen text. These are less comınonly employed as cryptanalytic techniques but are nevertheless possible avenues of attack.

Only relatively weak algorithms fail to withstand a ciphertext only attack. Generally, an encryption algoritbm is designed to withstand a known-plaintext attack.

Two more definitions are worthy of note. An encryption scheme is computationally secure if the ciphertext generated by the scheme meets one or both of the following criteria:

- The cost ofbreaking the cipher exceeds the value ofthe encrypted information.

- The time required to break the cipher exceeds the useful lifetime of the information.

The rub is that it is very difficult to estirrıate the aınount of effort required to crypt analyze ciphertext successfully, However, assuming there are no inherent mathematical weaknesses in the algorithm, then a bnıte-force approach is indicated, and here we can make some reasonable estimates aboutcöst:s and time.

A brute-force approach involves trying every possible key until an intelligible translation of the ciphertext Into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. Table 2.2 shows how much time is involved for various key sizes, The 56-bit key size is used with the DES {data encryption standard) algorithm. For each key size, the results are shown assumirrg that it takes 1 µs to perform a single decryption, which is a reasônable <>rde:riofrrıağnitııde for today's machines. With the use of massively parallel6:rganiza:tiôrls ôfmicroprocessors, it may be possible to achieve processing rates rriaııy ôtde:rs <>ffüağıfüude greater. The final column of Table 2.2 considers the results for a s)fstem. thaf can process 1 million keys per microsecond, As you can see, at this performance level, DES can no longer be considered computationally secure.

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/µs	Time Required at 10^6 Decryption/µs
32	$2^{32} = 4.3*10^{9}$	2^{31} µs= 35.8 minutes	2.15 milli seconds
56	$2^{56} = 7.2*10^{16}$	2^{55} µs= 1142 years	10 hours
128	$2^{128} = 3.4*10^{38}$	2^{127} µs= $5.4*10^{24}$ years	$5.4*10^{18}$ years
168	$2^{168} = 3.7*10^{50}$	2^{167} µs= $5.9*10^{36}$ years	$5.9*10^{30}$ years

Table 2.2 Average Time Required for Exhaustive Key Search

2.4 Private Key Encryption Algorithms

The most commonly used private key encryption algorithms are block ciphers. A block cipher processes the plaintext input in fixed-size blocks and produces a block of ciphertext of equal size for each plaintext block, The two most important private key algorithms, both of which are block ciphers, are the Data Encryption Standard (DES) and the Triple Data Encryption Algorithm (TDEA). We look at these two algorithms and then at the planned Advanced Encryption Standard (AES). We then provide a brief overview of other popular private key encryption algorithms.

2.4.1 Data Eneryption Standard

The most widely used encryption scheme is defined in the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). in 1994, NIST "reaffirmed" DES for federal use for another five years in FIPS PUB 46-2. The algorithm itself is referred to as the data encryption algorithm (DEA) [2].

The overall scheme for DES encryption is illustrated in Figure 2.2. The plaintext is 64 bits in length and the key is 56 bite in length; longer plaintext amounts are processed in 64-bit blocks,

The left-hand side of the figure shows that the processing of the plaintext proceeds in three phases. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input, This is followed by a phase consisting of 16 iterations of the same function. The output of the last (sixteenth) iteration consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the pre output. Finally, the pre output is passed through a permutation (IP-1) that is the inverse of the initial permutation function, toprodiıce the.ô-l-bit ciphertext,

The right-hand portion of Figµre 2.2 shows the way in which the 56-bit key is used. Initially, the key is passed throuğh aperın.utation function, Then, for each of the 16 iterations, a sub key (Ki) is produced by the conıbination ofa left circular shift and a permutation. The permutation function is the same for each iteration, but a different sub key is produced becaııse of the repeated shifting of the key bits.

The 64-bit permuted input passes through 16 iterations, producing an intermediate 64-bit value at the conclusion of each private key encryption algorithms iteration. The left and right halves of each 64bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (rights), The overall processing at each iteration can be summarized in the following formulas:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

Where $ denotes the bffwise :X.ORtru.nction.

Thus, the left-hand ôııtput ô:fitefatio:tl.(Ei)is si:rn.ply equal to the right-hand input to that iteration (Li-1). The right-hand outpüt' (Ri) is the exclusive-Ok of Li-1 and Ki. This complex function involves both permutation and substitution operations. The substitution operation, represented as tables called "Sıboxes", simply maps each combination of 48 input bits into a particular 32-bit pattem.

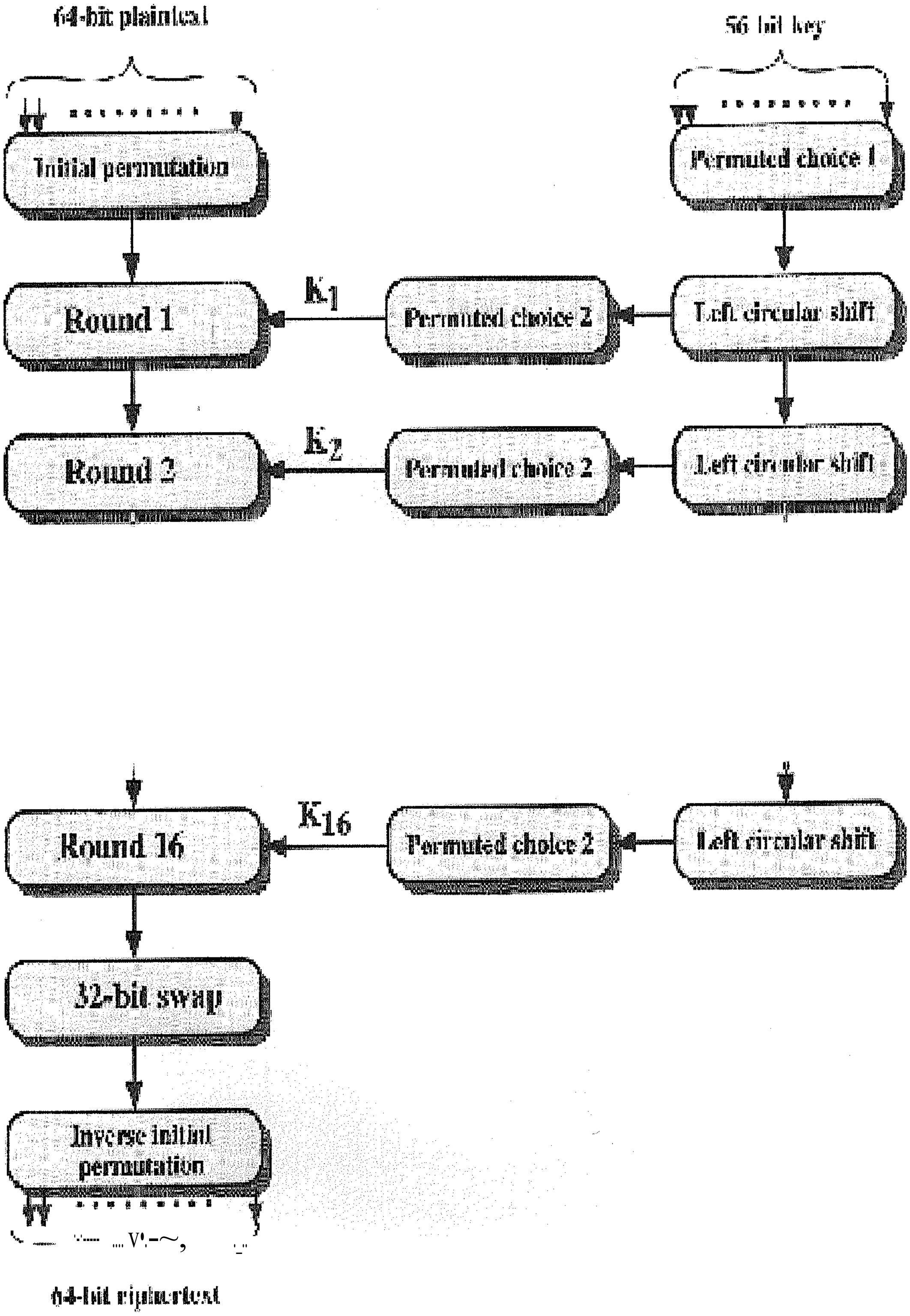


Figure 2.2 General Depiction of DES Encryption Algorithm

2.4.2 Triple Data Encryption Algorithm

Triple DEA (TDEA) was first proposed by Tuchman [TUCH79] and first standardized for use in financial applications in ANSI standard X9.17 in 1985. TDEA was incorporated as part of the data encryption standard in 1999, with the publication of FIPS PUB 46-3.

TDEA uses three keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence (Figure 2.6a):

$$C = E_{K3} [D_{K2} [E_{K1} [P]]]$$

Where

C = Ciphertext

P = Plaintext

EK[X] = encryption of X usingkey

DK[Y] = decryption of Y using key K

Decryption is simply the same operation withthekeys reversed (Figure 2.3b):

$$P = D_{K1} [E_{K2} [D_{K3} [C]]]$$

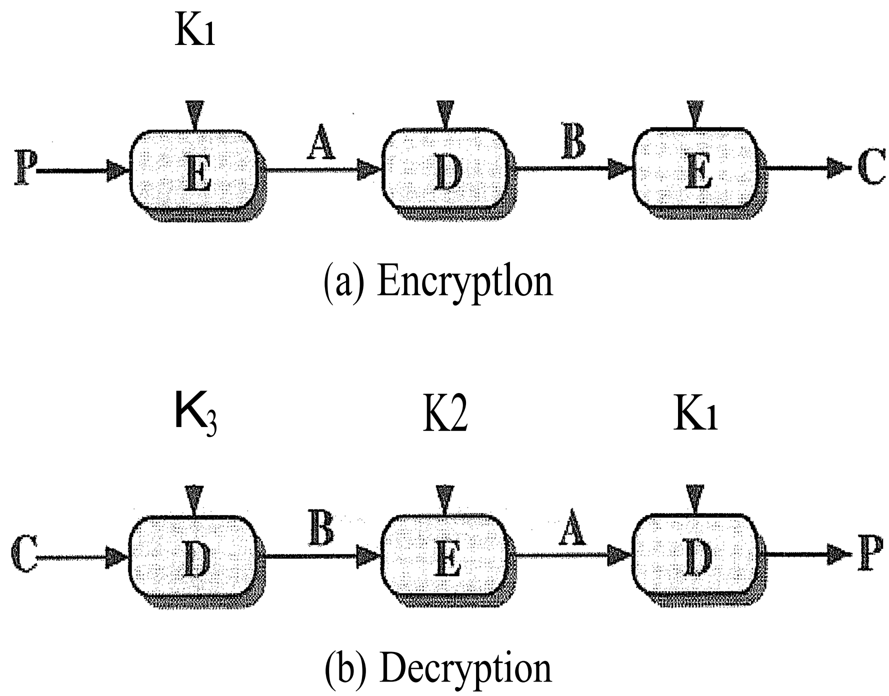


Figure 2.3 Triple DEA

There is no cryptographic significance to the use nfdecryption for the second stage of TDEA encryption. Its only advantage is that it allows users of TDEA to decrypt <lata encrypted by users ofthe older single DES:

$$C = E_{K1}[D_{K1}[E_{K1}[P]]] = E_{K1}[P]$$

With three distinct keys, TDEA has an effective key length of 168 bits. FIPS 46--3 also allows for the use oftwo keys, with K1 = K3; this provides fora key lerigth of 1 12 bits. FIPS 46-3 'inclıdesthe following guidelines for TDEA:

TDEA is the FIPS approved conventional encryption algorithm of choice.

- **The original DEA, which uses a single 56-bit key,** is permitted under the **standard for legacy systems only. New procurements** should support TDEA.

- Government organizations with legac)'OEA systems are encouraged to transition to TDEA.

- it is anticipated that TDEA and the advanced encryption standard (AES) will coexist as FIPS-approved algorithms, allowing for a gradual transition to AES.

it is easy to see that TDEA is a formidable algorithm. Because the underlying cryptographic algorithm is DEA, TDEA can claim the same resistance to cryptanalysis based on the algorithm as is claimed for DEA. Further, with a 168-bit key length, brute-force attacks are effectively impossible.
We can expect that TDEA will see increasing use over the next few years as the limitations of DES become intolerable and while waiting for the full-scale deployment ofAES.

2.4.3 Advaneed Encry.ptionStandard

TDEA has two attractions thatass~e itj"7idespread use over the next few years. First, with its 168-bit key length, it overc<>n:ıeş.the Vlı1.1lerability to brute-force attack of DEA. Second, the underlying encryptiortialgoritlını.dnTDEA is the same as in DEA.

This algorithm has been subjected to mote scrutiny than any other encryption algorithm over a longer period of time, and no effective cryptanalytic attack based on the algorithm rather than brute force has been foıınd. Accordingly, there is a high level of confidence that TDEA is very resistant to cryptanalysis. If security were the only consideration, then TDEA would be an appropriate choice for a standardized.
The principal drawback of TDEA is that the algorithm is relatively sluggish in software. The original DEA was. designed for mid-1970s hardwarç.jınplementation and does not produce efficientso:ftwareppde. TDEA, whichhas.thteetiınesas many rounds as DEA, is correspondingly Slô'\\l'~r. .A $~ç9ndarYcJrawp~ç!(isthat hoth DEA and TDEA use a 64-bit block size. For reas()ı:ıs•.()f1,ğtlı..çfficiç11çy<a1 bsecurity, a larger block size is desirable.
Because ofthese drawbacks, TDEAisrıofateasonable candidate for long term use, As a replacement, NIST in 1997 issued a call.for proposals for a new advanced encryption standard (AES), which should have a security strength equal to or better than TDEA and significantly improved efficiency. In addition to these general requirements, NIST

specified that AES must be a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits. Evaluation criteria include security, computational efficiency, memory requirements, hardware and software suitability, and flexibility.

In a first round of evaluation, 15 proposed algorithms were accepted. A second round narrowed the field to 5 algorithms. As of this writing, NIST hopes to complete the evaluation process and pick a fmal standard by the summer of 2001. Marketplace acceptance may take several years after that.

2.5 Oth.er Prlvate Key Algoritbms (Block Ciphers)

Rather than totally reinventing the wheel, virtually all contemporary private key block encryption algorithms use the basic Feistel block structııre. The reason is that this structnre is well understood and thls makes it easier to determine the cryptographic strength of a new algorithm. If an entirel:Y/different<structure were used, the new structure may have some subtle wea.knessnotiri:ımediately apparent to the designer. in this section we look at some of the other ciphers, in addition to DES and TDEA that have gained commercial acceptance.

2.5.1 Iaternatieaal Data Encryption Algorithm

The Intemational Data Encryption Algorithm (iDEA) is a syınmetric block cipher developed by Xuejia Lai and James Massey of the Swiss Federal Institute of Technologyin 1991 [LAI91]. iDEA uses a 128-bit key, lDEAdiffers.markedly from DES both in-the-round function.and in the sub key generatiön function, For the round function, iDEA .does >not >use. S-boxes. Rather, iDEA relies on three different mathematical operations: XOR, binary addition. of 16..bit integers, and binary multiplication of J6..bit iı:ı.tegers/T'J.ıesefunctiohs iare combined in such a way as to produce a complex transfornıatioft thatis vety difficult to analyze and hence very difficult to crypt analyze. The sub key genera.tfonalgorithm relies solely on the use of circular shlfts but uses these in a complex way to generate a total of six sub keys for each of the eight rounds ofIDEA.

Algorithm	Key Size	Number of Rounds	Mathematieal Operations	Applieaüons
DES	56 bits	16	XOR, fixed S boxes	SET, Kerberos
TripleDES	112 or 168 bits	48	XOR, fixed S boxes	Financial key management, PGP, S/MIME
iDEA	128 bits	8	XOR, addition, multiplication	PGP
Blowfish	Variable 'to 448 bits	16	XOR, variable, S-boxes, addition	
RC5	Variable to 2048 bits	Variableto 255	XOR, addition, subtraction, rôtation	
CAST-128	40 to 128 bits	16	XOR, addition, subttaction, rotation, fixed Sboxes	PGP

Table 2.3 Private Key Encryption Algorithm

Because IDEAwas one of the earliest of the proposed 128..füfreplacefuetıtsforDES, it has undergone corısidetable <scrutiny and so far appears tö be highly resistant to cryptanalysis. iDEA is usedi̇i PGP (as one alternative) and is also used in a number of commercial produôts,

2.5.2 Blowfish

Blowfish was developed in 1993 byBruce Schneier [SCHN93, SCHN94], an independent consultant and cryptographer, and quickly became one of the most popular alternatives to DES. Blowfish was designed to be easy to implement and to have a high execution speed. It is also a very compact algorithm that can run in less than 5K of memory. An interesting feature ofBlowfish is that the key length is variable and can be as long as 448 bits. In practice, 128-bit keys are used. Blowfish uses 16 rounds.

Blowfish uses S-boxes and the XOR function, as does DES, but also uses binary addition. Unlike DES, which uses fixed S-boxes, Blowfish uses dynamic S-boxes that are generated asa function of the key. In Blowfish, the sub keys and the S-boxes are generated by repeated application ofthe Blowfish algorithm itself to the key. A total of 521 executions of the Blowfish encryption algorithm are required to produce the sub keys and S-boxes. Accordingly, Blowfishis.not suitable for applications in which the seoret key changes frequently.

Blowfish is one oflhe)ınostforınidable conventional encryption algorithms so far implemented because both the sub keys and the S-boxes are produced by a process of repeated applications of Blowfishitself,(-whicfi.tlioroµğhly mangles the bits and makes cryptanalysis verydifficult. So far, therehaveih.~~:rı.a/few.published papers on Blowfish cryptanalysis, but no practical weaknesses have beeıifound.

Blowfish is used in a number of commercial applications.

2.5.3 RC5

RC5 was developed in 1994 by Ron Rivest [RIVE94, RIVE95], one of the inventors of the public-key algorithm RSA. RC5 is defined in RFC 2040 and was designed to have the following characteristics:

- Suitable für hardware or software: RC5 uses only primitive computational operations commonly found on microprocessors.

- Fasn To achieve this, .RC5 is a giınple algorithm and is word oriented, The basic operations work on fu.ll wortls of data at a time.

- Adaptable to proeessors of different word Iengthsı The mımber of bits in a word is a parameter of RC5; different word lengths yield different algorithms.

- Variable number of rouads; The number of rounds is a second parameter of RC5.This parameter allows a tradeoff between higher speed and higher security.

- Variable-length key: The key length is a third parameter of RC5. Again, this allows atradeoffbetween speed and security.

- Simpleı RC5's simple structure is easy to implement and eases the task of determininğ thtsttength of the algorithm.

- Low memery requirement: A low memory requirement makes RC5 suitable for smart cards ruıd other devices with restricted memory..

- High security; RC5 is intended to/pfövide high security with suitable parameters.

- Data-dependent reratiens: RCS incorporates rotations (eircular bit shifts) whose amount is data dependent. This appears to strengthen the algorithm against cryptanalysis

2.5.4 CAST -128

CAST is a design procedure for symmetric encryption algorithms developed in 1997 by Carlisle Adams and Stafford Tavares .ofEntrust Technologies [ADAM97]. üne specific algorithm developed as part ofthe CAST project is CAST -128, defined in RFC 2144, which makes use ofa key size that varies from 40 bits to 128 bits in S-bit incren

CAST is the result of a long process of research and development and has benefited from extensive review by cryptologists. It is beginning to be used in a number of products, including PGP.

CAST uses fixed Ssboxes, but ones that are considerably larger than those used in DES. These 8-boxes were carefully designed to be very nonlinear and resistant to cryptanalysis. The sub key-generation process used in CAST-128 is different from that employed in other conventional block encryption algorithms described in the literature, The CAST designers were concerned to make sub keys as resistant to known cryptanalytic attacks as possible and .felt that.the use of highly nonlinear 8-boxes to generate the sub keys from the main key provided this strength. Another interesting feature of CAST-128isthat the round function, F, differs from round to round, again adding to cryptanalytic:streııgth.

2.6 Cipher Block Môdes ofOperatfün

A symmetric block cipher processes 011.ebifhlockof data ata time. In the case of DEA and TDEA, the block length is 64 bits, For longer amounts of plaintext, it is necessary to break the plaintext into 64-bit blocks(padding the last block if necessary).

The simplest way to proceed is what is known as electronic codebook (ECB) mode, in which plaintext is handled 64 bits at a time and each block of plaintext is encrypted using the same key. The term codebook is used because, for a given key, there is a unique ciphertext for every 64-bit block of plaintext. Therefore, one can imagine a gigantlc. codebook in which there is an entry for every possible 64-bit plaintext pattern shöwiıığits ç6tresponding dphertext.
With ECB, ifthe sarn.e 64-bit'blijclcofplaintextappeats ınorethaıı once in the message, it always produces the sarn.e>cipli~#~xt, J3eçf:ltıseı ôf this, for Iengthy messages, the ECB mode maynot be secure.Ifthertı.essageisfüğhlystrucfüred, it may be possible fora cryptanalyst to exploit these regularitiesOFor exampfe, if it is known that the message always starts out with certain predefined fields, then the cryptanalyst may have a number of known plaintext-ciphertext pairs to work with. If the message has repetitive elements, with a period of repetition a multiple of 64 bits, then these elements can be

identified by the analyst. This may help in, the analysis or may provide an opportunity for substituting or rearranging blocks.

To overcome the security deficiencies of ECB, we would like a technique in which the same plaintext block, if repeated, produces different ciphertext blocks.

2.6.1 Cipher Bföck Cbaining Mode

In the cipher block chaining (CBC) mode (Figure 2.4), the input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block; the same key is used for each block. In effect, we have chained together the processing of the sequence of plaintext blocks. The input to the encryption function for each plaintext block bears rıo fixed relationship to the plaintext block. Therefore, repeating patterns of 64 bits are not exposed,

For decryption, each cipher blockis passedtht;ough the decryption algorithm. The result is XOR ed with the preceding ciphertextbfocktoproduce the plaintext block. To see that this works, we can write

$$C_1 = E_k [C_{1.1} \text{ EB } P_i]$$

Where EK[X] is the encryption of plaintext X using key K, and EB is the exclusive OR operation. Then,

$$D_k[C_i] = D_k [E_k (C_{1-1} \text{ EB } \mathbf{P_i)]}$$

$$D_k[C_i] = (C_{1.1} \text{ EB } P_i)$$

$$C_{1.1} \text{ EB } D_k [C_1] = C_{1-1} \text{ EB } C_{1-1} \text{ EB } \mathbf{P_i} = P_i$$

To produce the first block of ciphertext,aninitializatfon vector (IV) is XOR ed with the first block of plaintext. On decryptiorı, the IV is XOR ed with the output of the decryption algorithm to recover the first block of plaintext,

The IV must be known to both the sender and receiver. For maximum security, the IV should be protected as well as the key. This could be done by sending the IV using ECB encryption, üne reason for protecting the IV is as follows: If an opponent is able to fool the receiver into using a different value for IV, then the opponent is able to invert selected bitsin the first block ofplaintext, To see this, consider the following:

$$C_1 = E_K(IV \oplus P_1)$$

$$P_1 = IV \oplus D_K(C_1)$$

Now use the notation that $X[j]$ denotes the jth bit of the 64-bit quantity X. Then

$$\mathbf{P_1[i]} = IV\,\mathbf{[i]} \oplus D_K(C_1)\,\mathbf{[i]}$$

Then, using the properties ofXOR, we can state

$$\mathbf{P_1[i]'} = IV\,\boldsymbol{[i]} \oplus D_K(C_1)\,\mathbf{[i]'}$$

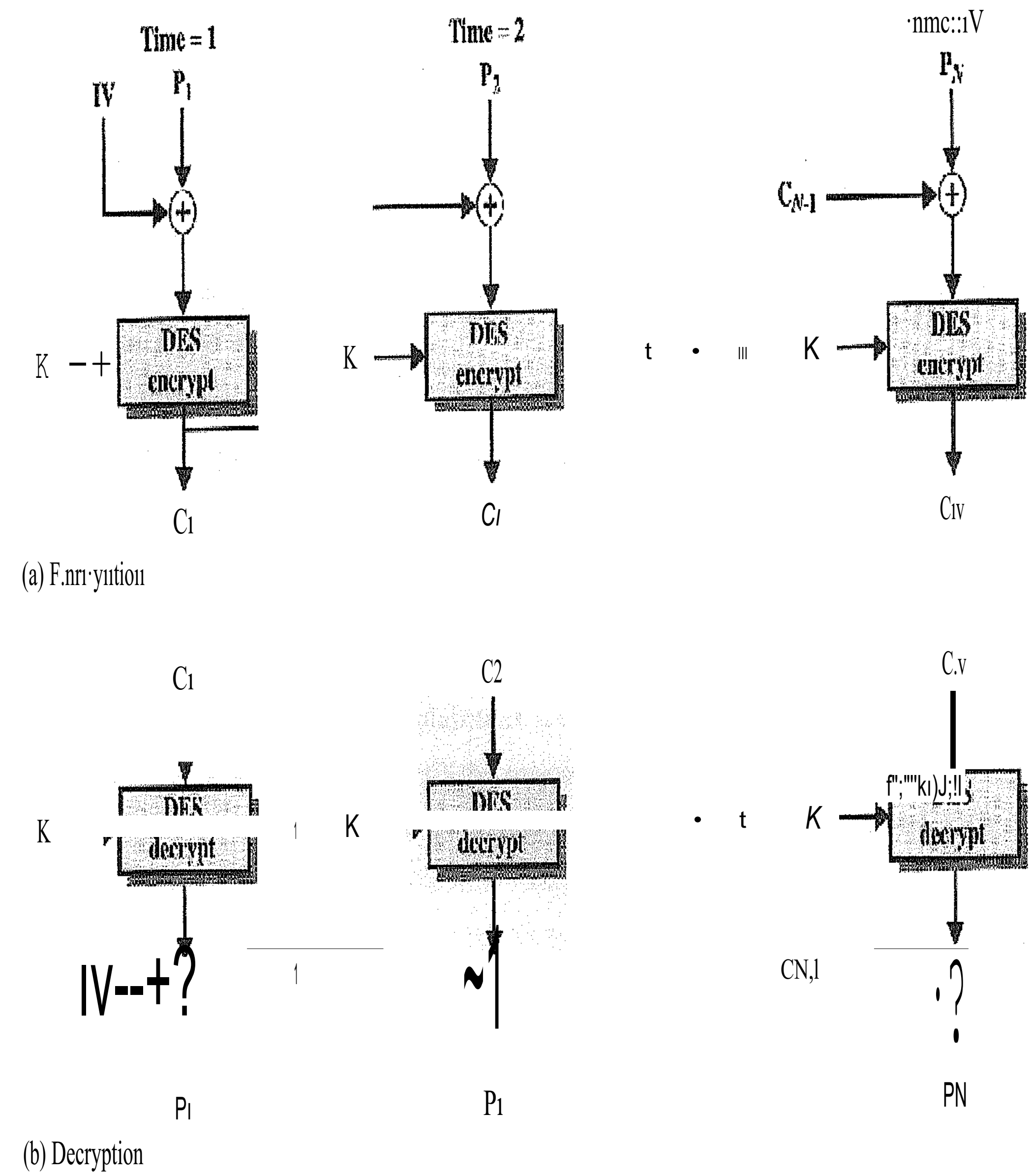


Figure *2A* Cipher Block Chaining (CBC) Motle

Where the prime rıotation denotes bit complementatiôiı; This means that ifan opponent can predictably change bits inIV, the correspondi:tıgbits oftheteceived value of Pi can be changed, CBC is widelyUsed..iııısecüritjrapplicatiôııs.

2.6.2 Cfpher Feedback Mode

The DES scheme is essentially a block cipher technique that uses 64-bit blocks. However, it is possible to convert DES into a stream cipher, using the cipher feedback (CFB) mode, A stream cipher eliminates the need to pad a message to be an integral number of blocks. It also can operate in real time. Thus, if a character stream is being transmitted, each character can, be encrypted and transmitted immediately using a character-oriented stream cipher.

üne desirable property of a stream cipher is that the ciphertext be of the same length as the plaintext. Thus, if 8-bit characters are being transmitted, each character should be encrypted using 8 bits. If more than 8 bits are used, transmission capacity is wasted.

We assume the CFBis assumed tha,tthe unit of transmission is j bits; a common value is j = 8. As with CBC, the units of plaintextare chained together, so that the ciphertext of any plaintext unit is a function ofallthe!prç~~dingplaintext.

First, consider encryption. The inputI~?~~e~1ryptfon function is a 64-bit shift register that is initially set to some initializatije"'f~t~r{IV).The leftmost (most significant) j bits ofthe output of the- encryption fu:rıctionareXOR ed with the first unit of plaintext P_1 to produce the first unit ofci~~e~f~tfl,.~~ichisthen transmitted. In addition, the contents of the shift register are sliifted.lçft l,y j bits and C_1 is placed in the rightmost (least significant) j bits of the shift register.J'fhis process continues until all plaintext units have been encrypted.

For decryption, except that the received ciphertext unit is XOR ed withthe output of the encryption function to produce the plaintext unit. Note that it is the encryption function that is used, not the decryption function. This is easily explained, Let Sj(X) be defined as the most significant j bits of X. Then

$$C_1 = P_1 \mathrm{EB} S[j](E(IV))$$

Therefore,

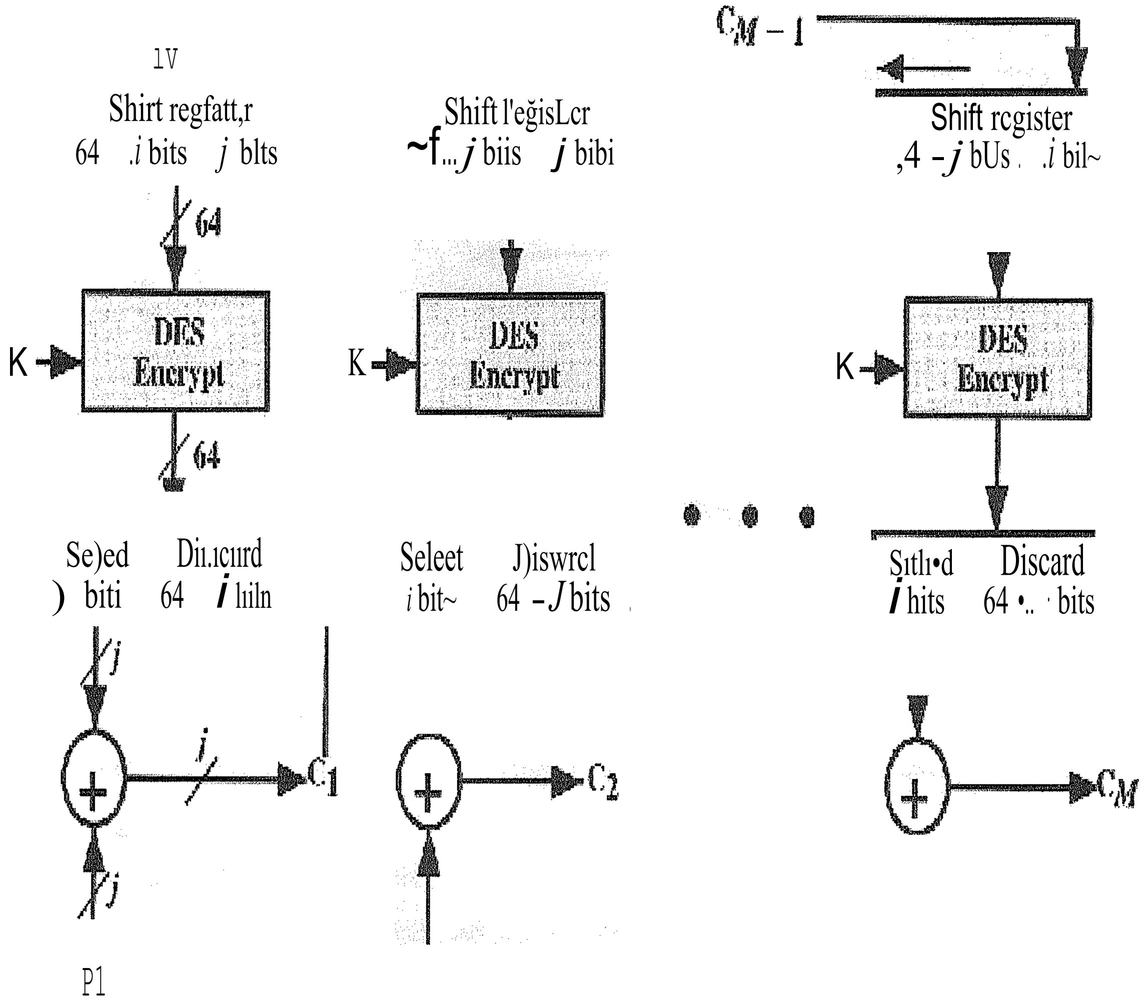
$$P_1 = C_1 \mathrm{EB} S[j](E(IV))$$

The same reasoning holds for subsequent steps in the process.

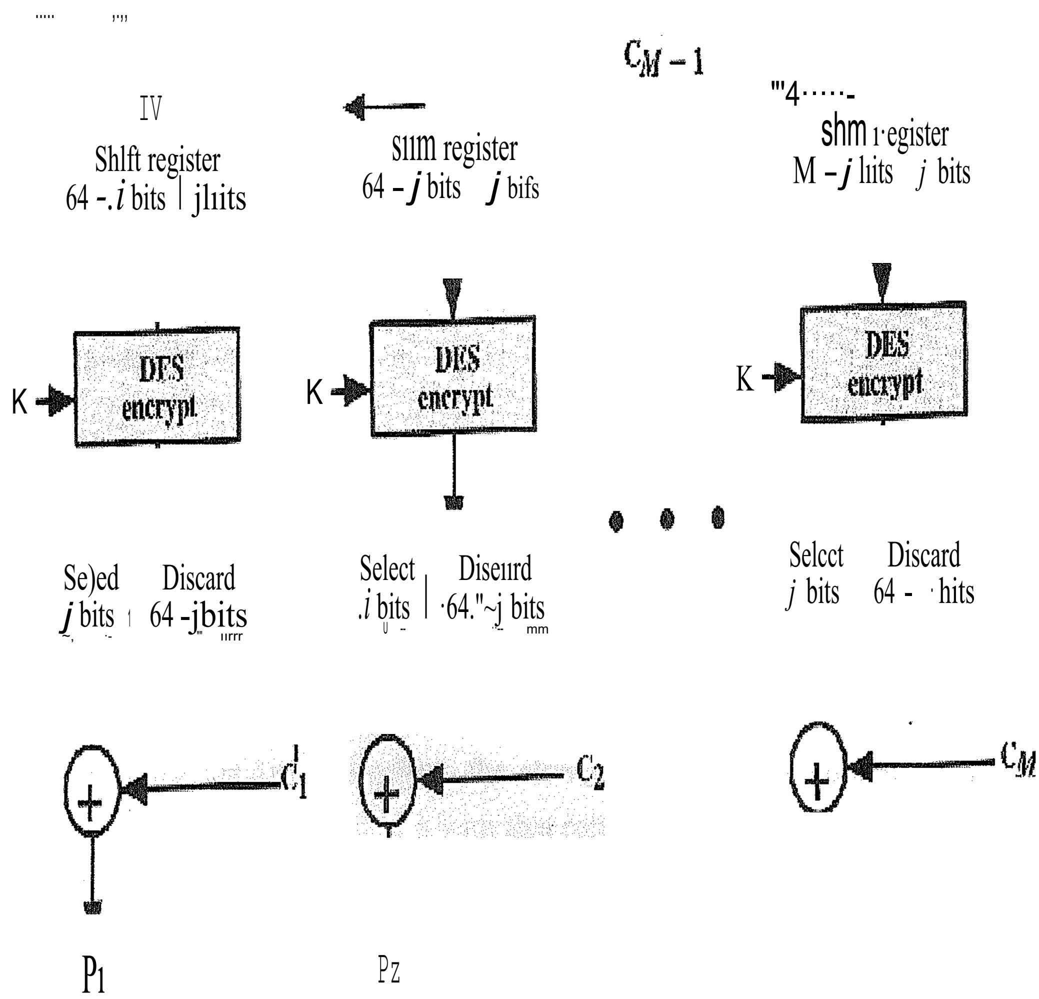
2.7 Leeation of Eneryptlo» Deviees

The most powerful, and most common, approach to countering the threats to network security is encryption, In using encryption, we need to decide what to encrypt and where the encryption gear should be located.

There are two fundamental alternatives:



(a) Rm:rn>tion



(b) Decryption

Figure 2.5 Cipher Feedback (CFB) Modes

Link encryption and end-to-end encryption; these are illustrated in use over a packet-switching network.

With link encryption, each vulnerable communications link is equipped on both ends with an encryption device. Thus all traffic over all communications links is secured. Although this requires a lot of encryption devices in a large network, it provides a high level of security. One disadvantage of this approach is that the message must be decrypted each time it enters a packet switch; this is necessary because the switch must read the address (virtual circuit number) in the packet header to route the packet. Thus, the message is vulnerable at each switch. If this is a public packet-switching network, the user has no control over the security of the nodes.

With end-to-end encryption, the encryption process is carried out at the two end systems. The source host or terminal encrypts the data. The data, in encrypted form, are then transmitted unaltered across the network to the destination terminal or host. The destination shares a key with the source and so is able to decrypt the data. This approach would seem to secure the transmission against attacks on the network links or switches. There is, however, still a weak spot.

2.8 Key Dlstribution

For private key encryption to work, the two parties to a secure exchange must have the same key, and that key must be protected from access by others. Furthermore, frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key. Therefore, the strength of any cryptographic system rests with the key distribution technique, a term that refers to the means of delivering a key to two parties that wish to exchange data, without allowing others to see the key.

Key distribution can be achieved in a number of ways for two parties A and B.

1. A key could be selected by A'and physically delivered to B.

2. A third party could select the key and physically deliver it to A and B.

3. If A and B have previously and recently used a key, one party côuldfra:nsmit the new key to the other, encrypted usingtheoldkey.

4. If A and B each have an encrypted connection to a third party C, C could deliver a key on the encrypted links to A and B.

Options 1 and 2 call for ına:nualdelivery of a key, For link encryption, this is a reasonable requirement, because each link encryption device is only going to be exchanging data with its partner on the other end of the link. However, for end-to-end encryption, manual delivery is awkward. In a distributed system, any given host or terminal may need to engage in exchanges with many other hosts and terminals over time. Thus, each device needs anumber ofkeys, supplied dynamically. The problem is especially difficult in a wide area distributed system.

Option 3 is a possibilityforeither link encryption or end-to-end encryption, but if an attacker ever succeeds in gaining access to one key, then all subsequent keys are revealed. Even'if frequentchanges are made to the link encryption keys, these should be done manually. To provide keys för end-to-end encryption, option 4 is preferable.

Figüre 2.3 illustrates an implementation that satisfies option 4 for end-to-end encryption. in the figure, link encryption is ignored. This can be added, or not, as required. For this scheme, two kinds of keys are identified:

- Session key: When two end systems {hosts, terminals, etc.) wish to communicate, they establish a logical connection (e.g., virtual circuit). For the duration of that logical connection, all user dara are encrypted with a one-time session key. At the conclusion of the session, or connection, the session key is destroyed.

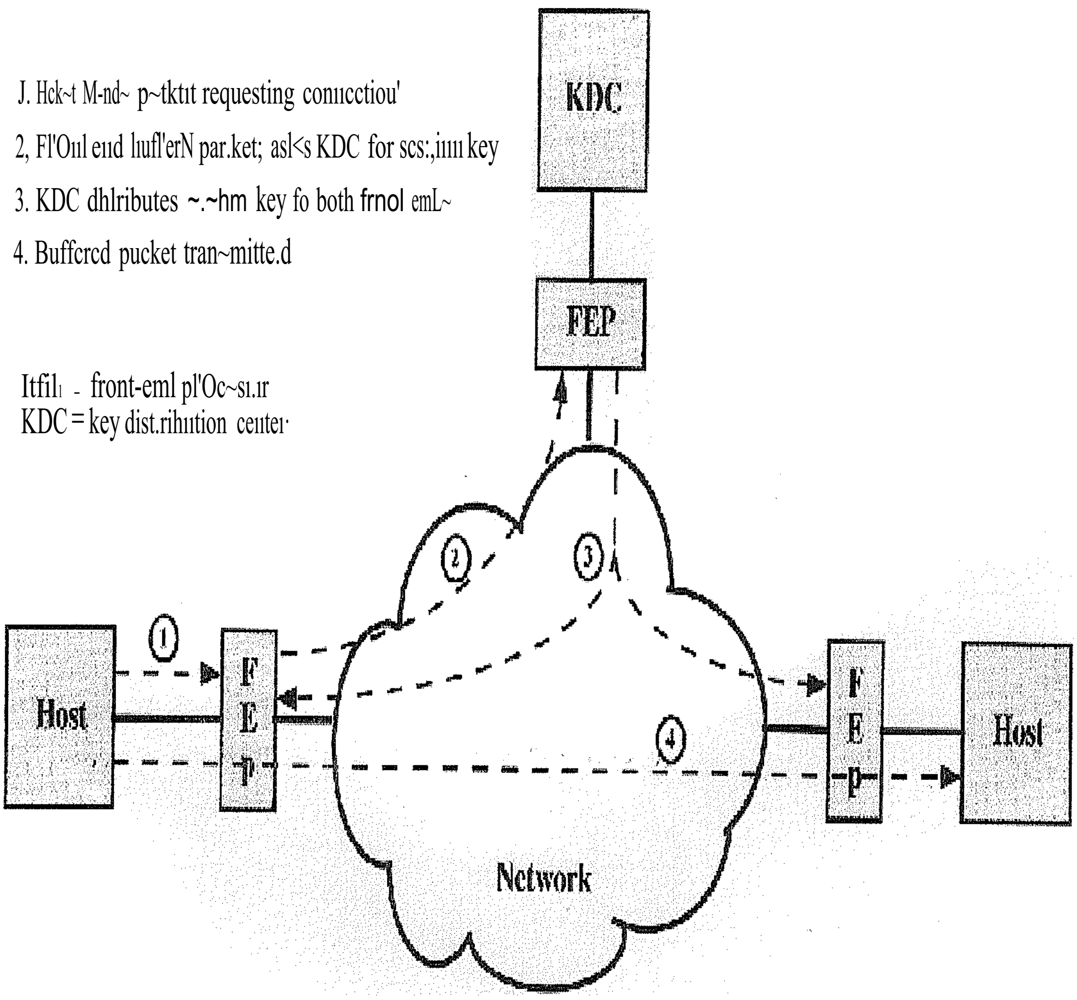
- Permanent .key: A permanent key is a .key used between entities for the purpose of distributing session keys.

The configuration consists ofthe following elements:

- Key distribution eenter: The key distribution center determines which systems are allowed to communicate with each other. When permission is granted for two systems to establish a connection, the key distribution center provides a one-time session key for that connection.

- Frnnt-end processor: *i* The frôııt-end processor performs end-to-end encryption and obtains sessioh keys 011 behalf of its host or terminal.

The steps involved in establishing a connection are shown in Figure 2.3. When one host wishes to set up a connection to another host, it transmits a connection-request packet (step 1). The front-end processor saves that pocket and applies to the KDC for permission to establish the connection {step .2). The communication between the FEP and the KDC is encrypted using a master key shared only by the FEP and the KDC. If the KDC approves the connection request, it generates the sessiOI! Key and delivers it to the two appropriate front-end processors, using a unique permanent key for each front end (step 3). The requesting front-end processor can now release the connection request packet, anda connection is set up between the two end systems (step 4). All user <lata exchanged between the two end systems are encrypted by their respective front-end processors using the one-time session key.



Ffgure 2.6 Automatic Key Distributions for Connection-Oriented Protocol

The automated key distribution approach provides the flexibility and dynamic characteristics needed to allow a number of terminal users to access a number of hosts and for the hosts to exchange data with each other.

CHAPTER3

PUBLIC-KEY CRYPTOGRAPHY

3.1 Publiç-1<:eYÇryptography Priaeiples

Public key cryptôgraphy was invented in 1976 by Whitfield Diffie and Martin Hellman. For this reasôn, it is sometime called Diffie-Hellman encryption, it is also called asymmetric encryption. A cryptographic systeın that uses two keys, a public key known to everyone and a private or secret key known only to the recipient of the message. An importaııfeleınent to the public key system is that the public and private keys are related in such a Way that only the public key can be used to encrypt messages and only the corresporıdlng private key can be used to decrypt them. Moreover; it is virtually impossible to deduce the private key if you know the public key.

Public-key systems, such as Pretty Good Privacy (PGP), are becoming popularfor transmitting information via the İnternet, They are extremely secure and relatively simple to use, The only difficulty with public-key systems is that you need tcrkn.ovvthe recipient's public key to encrypt a message for him or her. What's needed, therefore, is a global registry of public keys, which is one of the promises of the /fiew LDAP technology.

Of equal importance to private key encryption is public-key encryption, which finds use in message authentication and key distribution. This section looks first at the basic concept of public-key encryption and takes a preliminary look at key distribution issues. The two most important public-key algorithms: RSA and Diffie-Hellman that they introduce digital signature.

3.2 Publie-Key Encryptfon Strueture

Public-key encryption, first publicly proposed by Diffie and Hellman in 1976 [DIFF76], is the firsttfuly revolutionary advance in encryption in literally thousands of years. For one thin.ğ, public-key algorithms are based on mathematical functions rather than on siınple ôperations on bit pattems. More important, public-key cryptography is asymmetric, 'involving the use of two separate keys, in contrast to the symmetric private key encryption, which uses only one key. The use of two keys has profound consequences in the areas of confidentiality, key distribution, and authentication.

Before proceediňg, we should' first mention several common misconceptions concerning public•.key encryption. üne is that public-key encryption is more secure from cryptanalysis<thaı:ı priyatekey encryption. In fact; the security of any encryption scheme depends\hn(l)thelenğthôfthekey"aı:ıd(2)the côın.putational work involved in breaking a cipher. There is nothin.ğ in. priıciple a.bout either private key or public-key encryption that makes one superior' tö an.ôthef fröriı the point of view of resisting cryptanalysis. A second misconception is thatpublic-key encryption is a general-purpose technique that has made private key encryption obsolete. On the contrary, because of the computational overhead of current public-key encryption schemes, there seems no foreseeable likelihood that private key encryption will be abandoned.

Finally, there is a feeling that key distribution is trivial when using public...key encryption, compared to the rather cumbersome handshaking }involved with key distribution centers for private key encryption. In faet, seme föl'lıı. ôfPtötöcôl is needed, often involving a central agent, and the procedures inyolveda.re no sirrı.plef ot any more efficient than •thôse required for private•key en.ccyption..

A public-key encryption scheme has six ingredients (Figure 3.1):

- Plaintext: This is the reada.ble message or dara that is fed into the algorithn:ı as input.

- Eneryption Algorithm: The encryption algorithm perfonns various transfonnations on the plaintext.

- Pubhe and Private Key: this is a pair of keys that have been selected so that'if öne is used for encryption, the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input.

- Ctphertexe This is scrambled message produced as output. it depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertext,

Decryption algorithm: This algorithm accepts the ciphertext and the **matching key and produces the original plaintext**

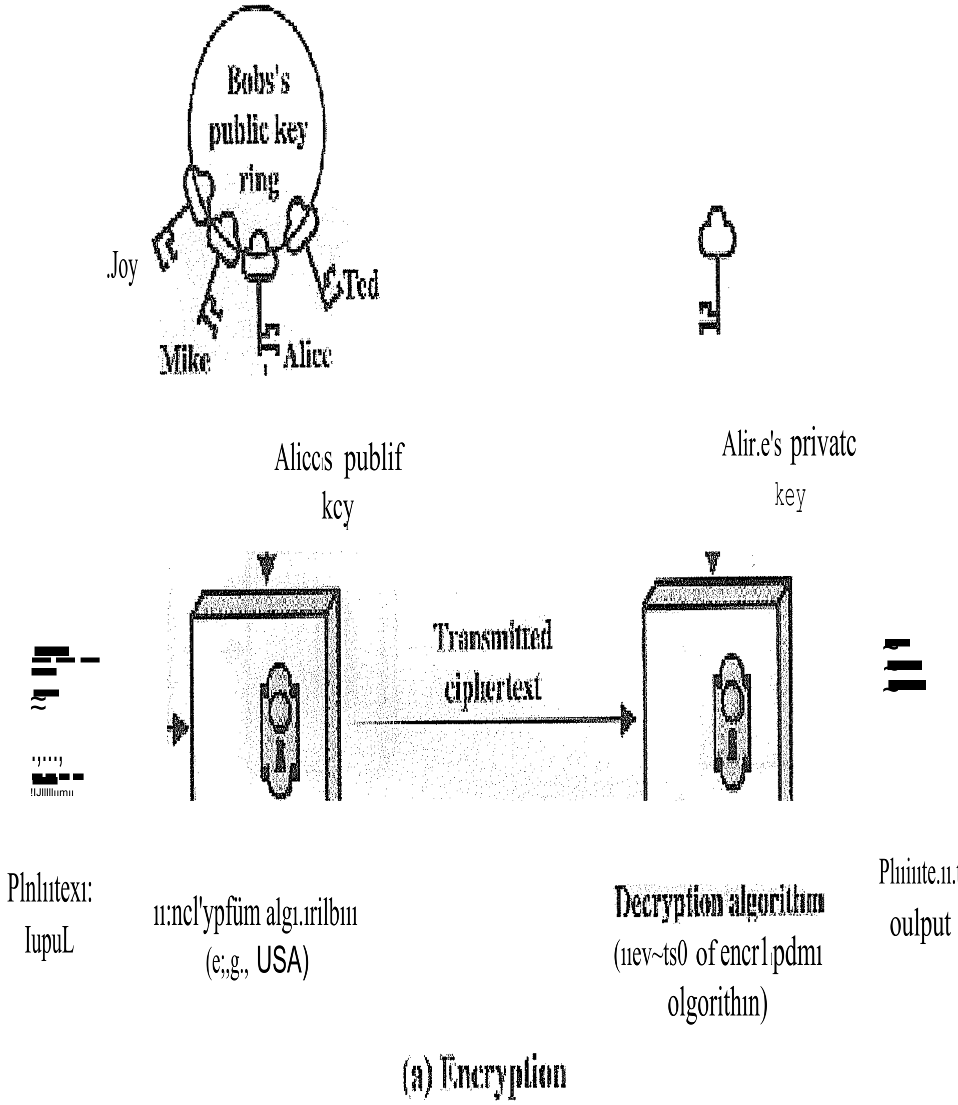


Figure 3.1(a) Public Key Cryptography for Encryption

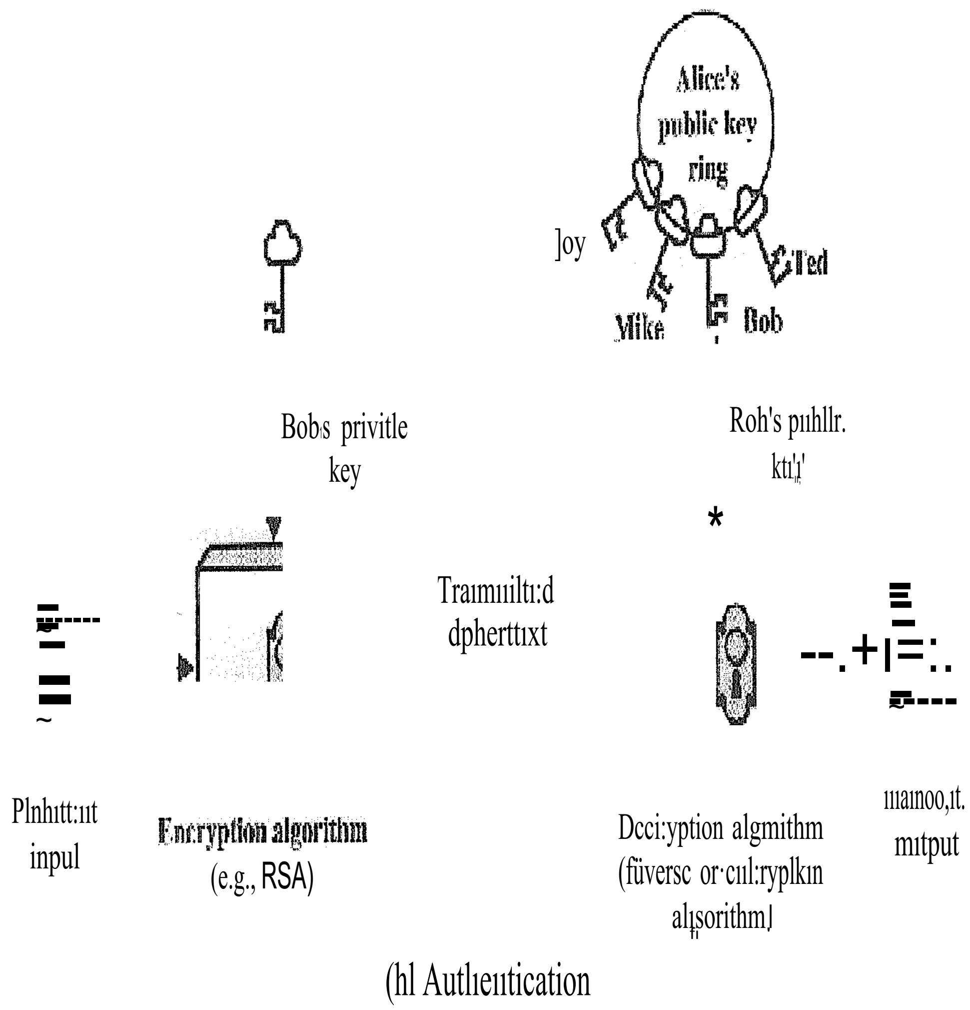


Figure 3.l(b) Public Key Cryptography for Authentication

As the names suggest, the public key of the pair is made public for otherstö Use, while the private key is known only to its owner. A general-purpose public-key cryptographic algorithm>telies on'one key for encryption and a different but related key for decryption. The esseııtia.1 steps are as follows:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.

2. Each user places one öffhetwôikeysin.a.ptıblfo register or other accessible file. This is the public key. The cempatiion keyisk.epfpriva.fo; As Figure 3.1a suggests, each user maintains a collection of public keys öbtainedfrôm others.

3. If Bob wishes to send a private message to Alice, Bob encrypts the message using Alice's public key.

4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

With this approach, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a üser protects his or her private key, incoming communication is seoure. At any time, a user can change the private key and publish the companion public key to replace the old public key.

The key used in .priya.te key encryption is typically referred to as a secret key. The two keys used for public-key eıicryption are referred to as the public key and the private key. Invariably, the privatekeyiskeptgecret.,butit.isreferred to asa private key rather than a secret key to avoid ponfusion witfrprivate key eııcryption.

3.3 Appllcations for Publie-Key Cryptosystems

Before proceeding, we need to clarify one aspect of public-key cryptosystems that is otherwise likely to lead to confusion, Public-key systems are characterized by the use ofa cryptographic type of algorithm with two keys, one held private aııd. one ayailable publicly, Depending on the application, the sender uses either the .serı.det's privatekey or the receiver's public key, or both, to perfonn some İ)'pe ()t~typt~~apltlc. functfon. In broad terms, we can classifythe use ofpublic-key cryptosystemsintothree categories:

Encryption-Decryption: The sender encrypts a messaze with the recipient's **public key**.

- Digital slgaature; The Seııder ıı.signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

- Key exehaage, Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parti es.

Algoritlım	Encrypfüm/Decryption	Digital Siguahıre	l(ey Exchange
RSA	Yeıı	Yes	Ye~
Diffic-Hdlımm	No	No	Yes
nSS	No	Yes	No
Elliptic Curve	y.,.	Yes	yes

Table 3.1 Applications for Public-Key Cryptosystems

Some algorithms are suitable for allthree aı,plications, whereas others can be used only for one or two of these applicatioııs. Table.3.liıidicates the applications supported by the algorithms discussed in this chapter, RSA and Diffie Hellman. The table also includes the digital signature standard (DSS) and elliptic-curve cryptography.

3.4 Requirements for Publie-Key Cryptography

The cryptosystem depends on a cryptographic algorithm based on two related keys. Diffie and Hellman postulated this system without demonstrating that such algorithms exist. However, they did lay out the conditions that such algorithms must fulfill [DIFF76]:

1. **It is computationally easy for a party B to generate a pair, (public key KU_b, private key KR_b).**

2. **It is computationally easy for a sender A,** lmowiııg the public key and the message to be encrypted, M, tö geıierate the corresponding ciphertext:

C=EKUb{M)

3. It is computationally easy for the receiver B to decrypt the resulting ciphertext:

C= DKRb(C)= DKRb[EKUb(M)]

4. It is computationally infeasible foran opponent, knowing the public key, KUb, to deterriiiıe 'the private key, KRb.

5. It is corrıputatiönally infeasible foran opponent, knowing.the public key, KUb, and a cipheitext, C, to recover the original message, M.,

We can adda sixth requ.irementthat, although useful, is not necessary for all public-key applications:

6. Either ofthe two related keys canpel.ısedfüt eııctyption, with the other used for decryption.

M= DK.Rb [EKUb(M)]= DKUb [EKUb(M)]=DKUb[EKRb{M)]

3.5 Puhlic-Key Ccyptography Algorithms

The.\two most widely used publio-key algorithms are RSA and Diffie-Hellman We look atbôth ofthese in this section and then briefly introduce two other algorithms.

3.5.1 The RSA Public-Key Encryption Algorithm

One of the first public-key schemes was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in [RIVE78]. The RSA scheme has since that time reigned supreme as the most widely accepted and implemented approach to public-key encryption. RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and n - 1 for some n.

Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C:

$$C = M^{e} \bmod n$$

$$M = C^{d} \bmod n = (M)^{ed} \bmod n = M^{ed} \bmod n$$

Both sender and receiver must know the values of n and e, and only the receiver knows the value of d. This is a public-key encryption algorithm with a public key of KU = $\{e, n\}$ anda private key of KR = $\{d, n\}$. For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

1. It is possible to fin.dValuesof e, d, n such that $M = M^{ed}$ mod n for ali $M < n$.

2. It is relatively easy to calculate M^{e} and C for all values of $M < n$.

3. it is infeasible to determine d ğiverte andn.

The first two requirements are easily met. The third requirement can be met for large values of e and n.

The RSA algorithm begins by selecting two prime numbers, p and q, and calculating their product n; which is the modulus for eneryption and dR~rypti-on. Next, we need the quantity $\phi(n)$, referred to as the Euler totient of n; which is the number of positive integersless than n and relatively prime to n. Then select an integer e that is relatively prime to $\phi(n)$ [i.e., the greatest common divisor of e and $\phi(n)$ is 1]. Finally, calculate d as the mul'tip.li:vat've inverse of e, modulo $\phi(n)$. It can be shown that d and e have the desired prop''rlies.

Suppose that user A has published its public key and that user B wishes to send the message M to A.

Then B calculates C = M^{e} (mod n) and füınsfuits C. On receipt of this ciphertext, user A decrypts by calculating;

$$M = C^{d} \pmod{n}.$$

An example is shown; the keys were generated as follows:

1. Select two prime numbers, $p = 7$ and $q = 17$.
2. Calculate $n = pq = 7 \times 17 = 119$.
3. Calculate $\varphi(n) = (p-1)(q-1) = 96$.
4. Select e such that e is relatively prime to $\varphi(n) = 96$ and less than $\varphi(n)$; in this case e=5

Key Gene.ration

Select p,q	p and q bothprirne
Calculate n- p*q	
Calculate cp(n)-(p-1) (q-1}	
Select integer e	gcd(cp(n),e) = 1; 1< e<
Calculate d	d = e-11110d cp(n)
Public key	KU=·{e,
Private key	KR = {d,n}

E:ncryption

Plaintext $M<n$

Ciphertext $C= M^{e} \pmod{n}$

Decryption

Ciphertext

Plaintext $M = C^{d} \pmod{n}$

Figure 3.2 The RSA Algorithm

5. Determine d such that $de = 1 \bmod 96$ and $d < 96$. The correct value is $d = 77$,. because $77 \times 5 = 385 = 4 \times 96 + 1$.

The resulting keys are public key KU = {5, 119} and private key KR = {77, 119}. The example shows the use of these keys for a plain text input of $M = 19$. For ericryption, 19 is raised to the fifth power, yielding 2476099. Upon division by 119, the remainder is determined to be 66. Hence, $19^5 = 66$ mod 119, and theciphertext is 66. For decryption, it is determined that $66^{77} = 19$ mod 119.

There are two possible' approaches to·· defeatingthe··RsA algorithm.' The firsfis the brute-force approach: Try ali pôssible private keys. Thus; the Iarger the nurnbet ofbits ine and *d,* them.ore securethealgorithm..

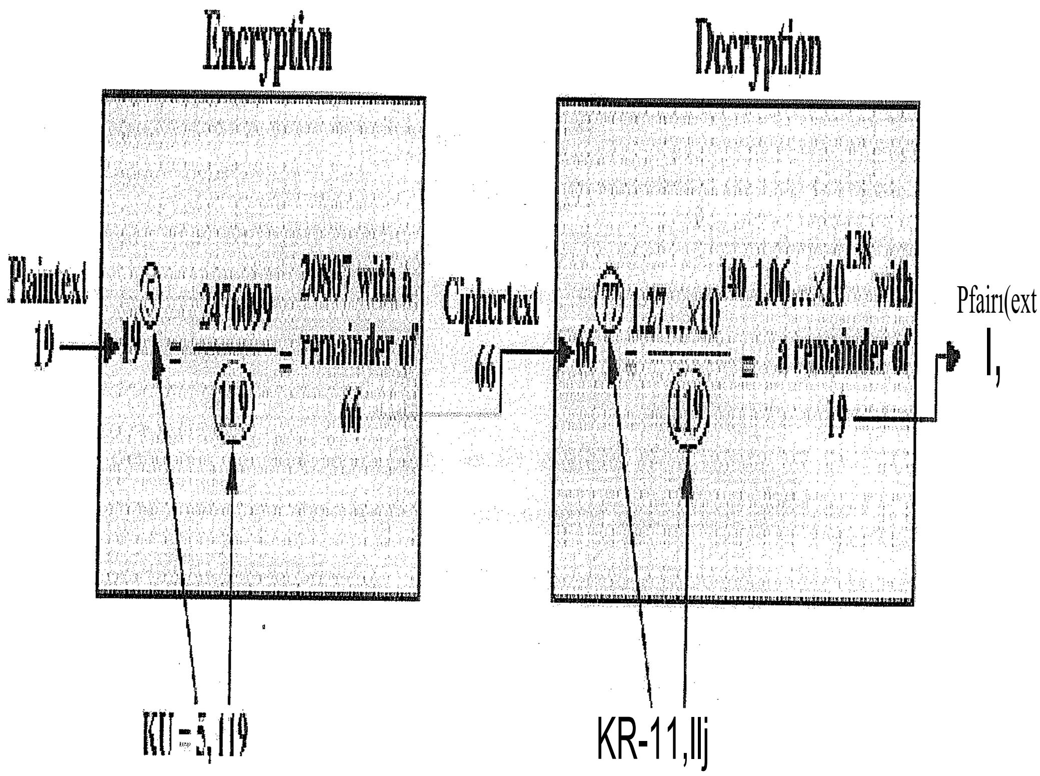


Figure 3.3 Example of RSA Algorithm

However, because calculations involved, both in key generation and in encryption/decryption, are complex, the larger the size ofthe key, the slower the system will run.

Most discussions of the cryptanalysis of RSA have focused on the task of facto ring n into its two prime factors. Fora large n with large prime factors, factoring is a hard problem, but not as hard as it used to be. A striking illustration of this is the following. in 1977, the three inventors of RSA dared Scientific American readers to decode a cipher they printed in Martin Gardner's "Mathematical Games" column [GARD77].

They offered a $100 reward for the return of a plain text sentence, an event they predicted might not occur for some 40 quadrillion years. in April of 1994, a group working over the Internet and using over 1600 computers claimed the prize after only eight months of work [LEUT94]. This challenge used a public-key size (length *of n*) of 129 decimal digits, or around 428 bits. This result does not invalidate the use of RSA; it simply means that larger key sizes must be used. Currently, a 1024-bit key size (about 300 decimal digits) is considered strong enough for virtually all applications,

3.5.2 Diffie-Helman Key Exchange

The first published public-key algorithm appeared in the seminal paper by Diffie and Hellman that defined public key cryptography[DIFF76] and is generally referred to as Diffie-Helman key exchange. A number of commercial products employ this key exchange technique,

The purpose of the algorithm **is to enable two users** to exchange a secret key securely that can then be used for subsequent **encrption** of messages. The algorithm itself is limited to the exchange of the keys.

The Diffie-Helman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.. Briefly, we can define the discrete logarithm in the following way. First, we define a primitive root of a prime number p as one whose powers generate all the Integers from 1 to $p-1$. that is.if a is primitive root of the prime number p, then the numbers

$$a \bmod p,\ a^2 \bmod p,\ \ldots,\ a^{p-1} \bmod p$$

are distinct and consist of the integers from 1 through p-1 in some permutation

3.6 Other Public-Key Cryptography Algorithms

Two other public-key algorithms have found commercial acceptance: Digital Signature Standard(DSS) and Elliptic-Curve Cryptography.

3.6.1 Digital Sigıfature Standard

The National-lnstitıte of Standards and Technology (NIST) has published Federal Information Processing Standard FIPS PUB 186, known as the digital signature standard(DSS). The DSS rnakes use oftheSHA-1 and presents a new digital signature technique, the digital signature algorithm {DSA).

The DSS was originally proposed in 1991 and revised in 1993 in response to public feedback concerningithe security of the scheme. There was a further mirıor revision in 1996. The DSS ııses aıı alğorithrn that is designed to provide only the digital signature function. Unlike RSA,ifcafuıôtbe ıisedfor encryption or key exchange,

3.6.2 Elhptle-Curve Cryptography

The vast majority of the products and standards that use public-key cryptography for encryption and digital signatures use RSA. The bit Iength for secure RSA use .has increased over recent years, and this has put a heavier processing load on applications using RSA. This burden has ramifications, especially for electronic commerce sites that conduct Iarge.numbers of secure transactions. Recently, a competing system has begun to challenge RSA: elliptic-curve oryptography (E =Already, ECC .is.showing up in standardization efforts, including the IEEE Pl363 Standard, for Public-Key Cryptography.

The principal attraction of ECC compared to RSA is that it appears to equal secıırity for a far smaller bit size, thereby reducing processing overheat. On the other hand, although the theory of ECC has been around for some time only recently that products have begun to appear and that there has been sust cryptanalytic interest in probing for weaknesses. Thus, the con:fidencelevel in *ECC* is not yet as high as that in RSA.

ECC is fundamentally more difficult to explain than either RSA or Diffie-Hellman, and a full mathematical description is beyond the scope of this book. The nique is based on the use ofa mathematical construct known as the elliptic curve.

3.7 Digital Signatures

Public-key encryption can be used in another way, as illustrated in Figure 3.lb. SupposethafBob wants to send a message to Alice and, although it is not important that the message be kept secret, he wants Alice to be certain that the message is indeed from him. In .this case Bob uses his own private key to encrypt the message, When Alice receives the ciphertext, she finds that she can decrypt it with Bob's public key, thus proving that the message must .have been encrypted by Bob. No one else has Bob's private key and therefore no one else could have created a ciphertext that could be decrypted with Bob's public key.\Therefore, the entire encrypted message serves as a digital signature. In addition, it is impossible to alter the message without access to Bob's private key, so the message is authe:nticatedboth in terms of source and in terms of data integrity.

In the preceding scheme, the entire message is encrypted which, although validating both author and contents, requires a great deal of störage, Each document must be kept in plaintext to be used for practical purposes: A copy also must be stored.in ciphertext so that the origin and contents can be verified in case ofa dispute. A more efficient way of achieving the same results is to encrypt a small block of bits that is a function of the document. Such.a. block, called an authenticator, must have the property that it is infeasible to change the document without changing the authenticator. If the authenticator is encrypted with the sender's private key, it serves as a signature that verifies origin, content, and sequencing. A secure hash code such as SHA-l can.serve this function.

It is important to emphasize that the encryption process just described does not provide confidentiality. .That is, the message being sent is safe from alteration but not safe from eavesdropping. This is obvious in the case ofa signature based on a portion of the message, because the rest of the message is transmitted in the clear. Even in the case of complete encryption, there is no protection of confidentiality becaııseany.observer can decrypt the message by using the sender's public key.

3.8 Key Management

üne of the ma'.jor roles of public-key encryption is to address the problem of key distributio:ıLThere Eıre actually two distinct aspects to the use of public-key encryption in this tegard:

- The distribution ofpublic keys
- The use ofpublic-key encryption to distribute secret keys

We examine each ofthese areas 'in turn.

3.8.1 Digifal Certifleates

On the face of it, the point ofpublic-key ehcryption is that the public key is public. Thus, if there is some broadly accepted ptiblic..key algorithm, such as RSA, any participant can send his or her public key to ruıy öther participant or broadcast the key to the community at large, Although this approach is convenient, it hasa major weakness:. Anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key. Until such time as user A discovers the forgery and alerts other participants, the forger is able to read all eıicrypted ınessağes intended for A and can use the forged keys for authentication.

The solution to Jfüs prôble~i0'.i1he pıiblic-~e~./cerlificate. In essence, a certificate consists ofa publickeyplusa,0s?)I~of~~~eyowner, withthe whole block signed by a trusted third party. Typically, the .thirdpartyis a certificate authority (CA) that is trusted by the user community, such as a. gôverıiment agency or a financial institution. A user can present his or her public key to the authority in a secure manner and obtain a certificate. The user can then publish the certificate. Anyone needing this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature,

üne scheme has become universally accepted for formatting public-key certificates: the X.509 standard. X.509 certificates are used in most network security applications, includingIP security, secure sockets layer (SSL), secure electronic transactions (SET), and•S/MIME.

3.8.2 Public-Key Distribution of Seeret Keys

With private key encryption, a fundamental requirement for two parties to communicate securely is that they share.a secret key. Suppose Bob wants to create a messaging application that will enable him to exchange e-mail securely with anyone who has access to the Internet or to some other network that the two of them share. Suppose Bob wants to do this using private key encryption. With private key encryption, Bob and his correspondent, say, Alice must eome up with a way to share a unique secret key that no one else knows. How are they going to do that? If Alice is in the next room from Bob, Bob could generate a key and write it down on a piece of paper or store it on a diskette and hand it to Alice.

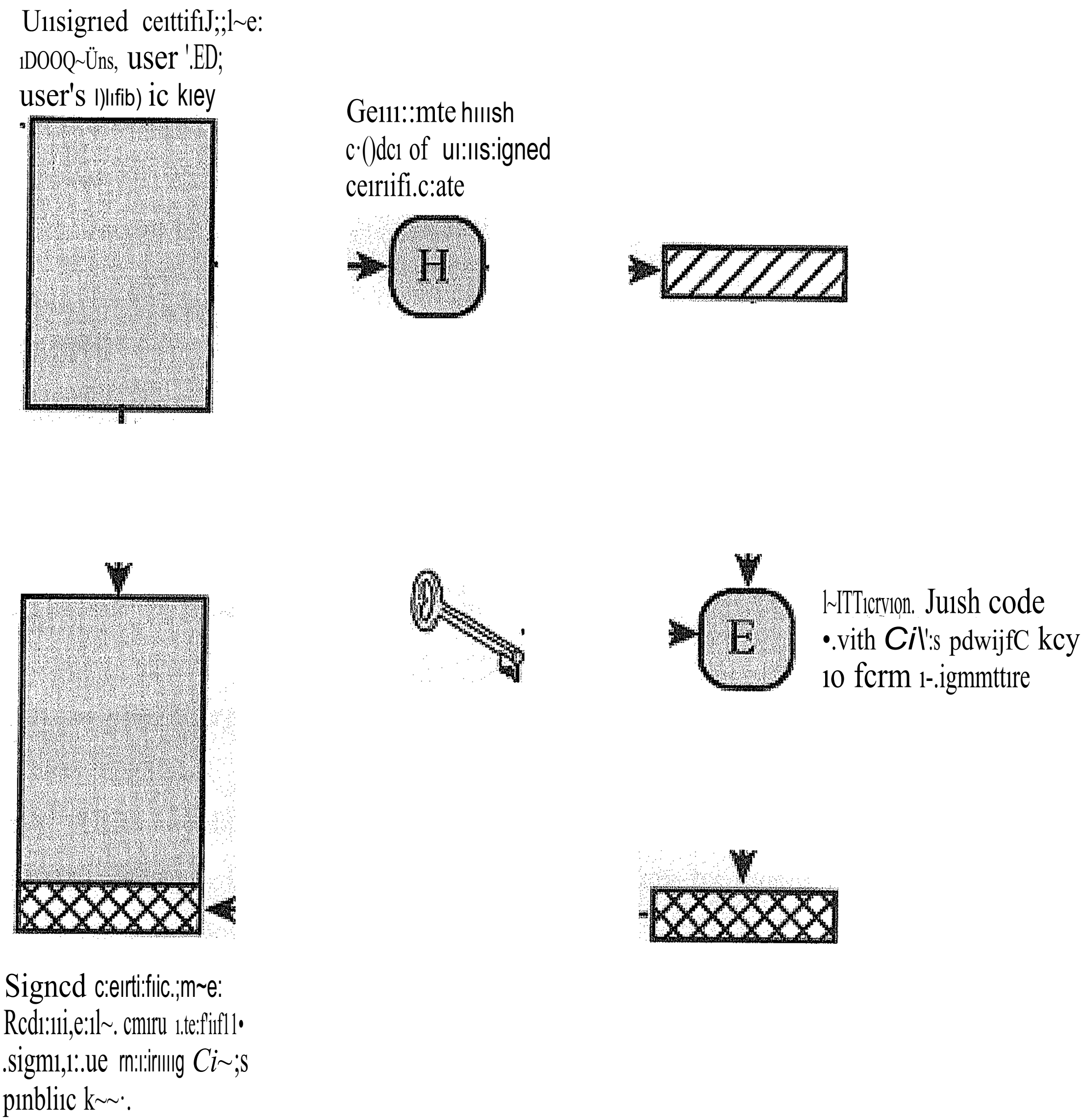


Figure 3.4 Public-Key CertifioateUse

But if Alice is on the other side of the continent or the world, what can Bob do? He **could encrypt this key using conventional encryption and** e-rnail it to Alice, but this means that **Bob and Alice must share a secret key to** encrypt this new secret key. Furthermore, Bob artd everyôrie else Whô usesthls new e-mail package face the same problem with every potential côıespondent: Each pair of correspondents must share a unique secret key.

üne approach is the use of Diffie-Hellman key exchange. This approachds indeed widely used. However, it suffers the drawback that, in its simplest form; Diffie-Hellman provides no authentication ofthe two communicating partners.
A powerful altemative is the use of public-key certificates. When Bob wishes to communicate with Alice, Bob can do the following:

1. Prepare a message.

2. Encrypt that message ıısing conventional encryption with a one-time conventional session key,

3. Encrypt the session key using public-key encryption with Alice's public key.

4. Attach the encrypted session key to the message and send it to Alice.

Only Alice capable of decryption the session key and therefore of recovering the original message. If Bob obtained Alice's public key by means of Alice's public-key certificate, then Bob is assured that is a valid key.

CONCLUSION

in this project, it has given a detailed view on the private key and public key cryptography and their applications and algorithms.

in computer security, cryptography is one of several ways of achieving isolation and is not one of the more important ways. in network security, cryptography is the main attraction everything is done via message passing (ultimately), so the only way to achieve confidentiality and the authentication needed for access control is through cryptography.

REFERENCES

[1J Symmetric cryptographic system for data encryption by C. ADAMS in Apr 1996.

[2] Cryptography and Network Security Principles and Practice, Second Edition, by William Stallings.

[3] Bruce Schneider, Applied Cryptography, Second Edition, John Wiley& Sons, ine, 758, 1996-1999.

[4] Random sources for cryptographic systems by G.B. AGNEW in 1988.

[5] Handbook of AppliedCryptography by A. Menezes, P.Van Oorschot, and S. Vanstone in 1996.