CRYPTOGRAPHY AND NETWORK SECURITY

Driving Growth and Success in the Digital Age

Cryptography

and

Network Security

Cryptography and Network Security

—----------

©

—----------

Published By:

(Publisher and Distributor)

Mob:

Email:

Web:

Year:

ISBN:

₹

Printed by:

—----------

Author's Introduction: Dr. Kamal Pandey

—----------

-

Dr. Kamal Pandey is a distinguished academician and scholar currently serving as an Assistant Professor in the Department of Sociology at Verma Shyamdulari P.G. College. With over a decade of rich experience in the field of education, he has made significant contributions to academia and research.

Dr. Pandey holds a Master's degree and a Ph.D. in Sociology, and his academic journey is marked by excellence and dedication. He has published his research in seven renowned international journals and has actively participated in 12 national and international conferences, enriching the discourse on education and sociology.

As a member of the editorial boards of two prestigious research journals, Dr. Pandey plays a key role in advancing scholarly research. He also serves as the Managing Director of the *Writers Crew International Research Journal*, where his leadership and insight are widely acknowledged. His outstanding contributions have earned him recognition and honors at the international level.

In addition to his academic pursuits, Dr. Pandey frequently serves as a resource person for research initiatives and is deeply involved in community service through NGOs. He actively works to empower individuals, fostering awareness and self-reliance.

Dr. Kamal Pandey is dedicated to strengthening the relationship between education and society. His visionary ideas and groundbreaking research continue to serve as a cornerstone for the development of education and social understanding.

Table of contents

Preface

In this age of viruses and hackers, of electronic eavesdropping and electronic fraud, security is paramount. As the disciplines of cryptography and network security have matured, more practical, readily available applications have evolved to implement network security. This

book provides a practical survey of both the principles and practice of cryptography and network security.

In this age of universal electronic connectivity, of viruses and hackers, of electronic eavesdropping and electronic fraud, there is virtually no time at which security does not matter. Two trends have come to make the subject of this book of vital interest. First, the explosive growth in computer systems and their interconnections through the network has increased the dependence of both organizations and individuals on the information stored and transmitted using these systems. This, in turn, has led to increased awareness of the need to protect data and resources from disclosure, to guarantee the authenticity of data and messages, and to protect systems from network-based attacks.

Second, the disciplines of cryptography and network security have matured, leading to the development of practical, readily available applications for implementing network security.

It is the purpose of this book to provide a practical survey of both the theory and practice of cryptography and network security. In the first part of the book, the basic issues addressed by a network security capability are explored by providing a tutorial and survey of cryptography and network security techniques. The latter part of the book deals with the practice of network security, the practical applications that have been implemented and are in use to provide network security. An attempt has been made to make the book self-contained. The book not only presents essential mathematical results, but also provides the reader with an intuitive understanding of those results.

The book is meant for both an academic and a professional reader. As a textbook, it may be useful for undergraduate/postgraduate students of Computer Science, Computer Engineering and Electrical Engineering as per their curriculum. The proposed book covers the basics of the science of cryptography. It explains how programmers and network professionals can use cryptography to maintain the confidentiality of computer data. Starting from the origin of cryptography, it explains cryptosystems, various traditional and modern ciphers, public key encryption, data integration, message authentication, and digital signatures. This book is meant for students of Computer Science who wish to learn the basics of cryptography. It will also be useful for network working professionals who wish to incorporate various cryptographic algorithms to ensure secure data communication on their network. Network security deals with all aspects related to the security of sensitive information assets present on the network. It includes various mechanisms developed to provide fundamental security services for data communication.

This book explains the details of various types of network vulnerabilities and security measures employed against the attacks. It describes the functioning of the most common security protocols employed on various network layers from application to data link layer. Through this book, students will gain intermediate level of knowledge about network security. This book will be helpful for beginners to understand the basics of network security. This book will be useful for those who are interested in taking up a career in the field of information and network security.

Chapter - 1

Cryptography - An Introduction

Origin of Cryptography

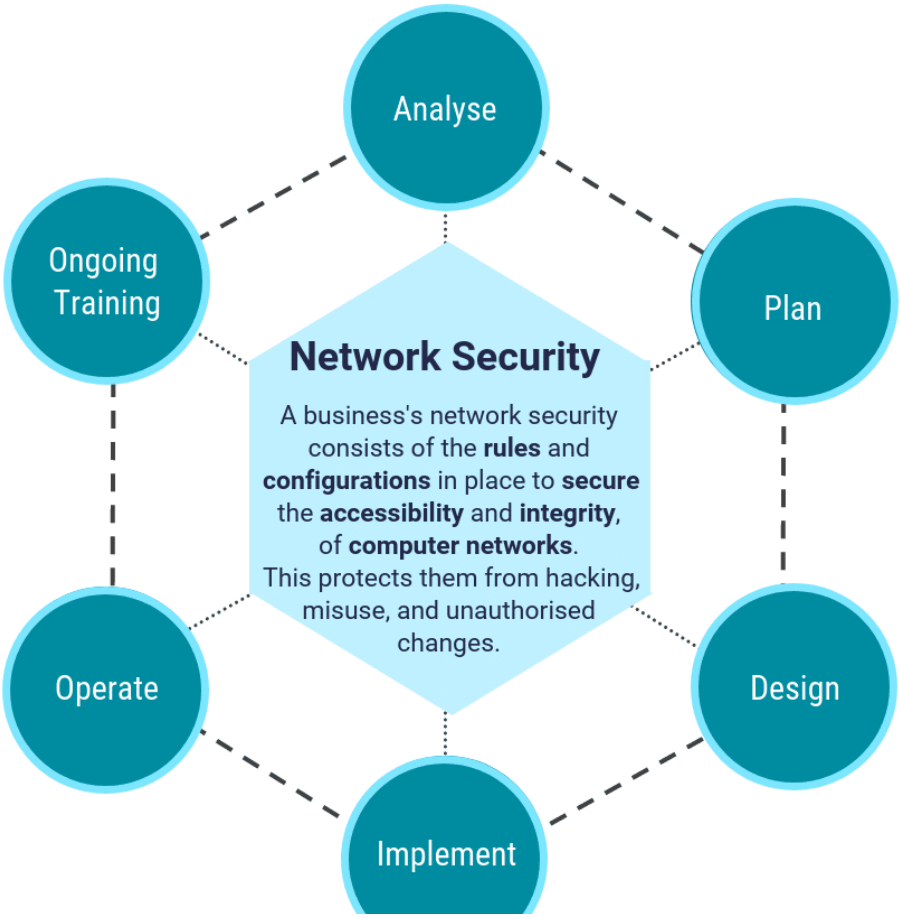
The word ‘cryptography’ was coined from the combination of two Greek words, ‘krypta’ meaning hidden and ‘graphein’ meaning writing.

Humans have always had two underlying needs -

(1) to communicate and share information and

(2) to communicate among selected individuals. Both these needs gave rise to the art of coding messages so that only the desired individuals could access the information. Unauthorized people were unable to extract any information even if the encrypted message reached them. The art and science of concealing messages in order to introduce secrecy in the security of information is identified as cryptography. A message in its original form is known as plain-text or clear-text or plain-text. The encrypted information is known as cipher-text or enciphered text. The process of creating cipher-text from plain-text is known as encryption or decryption. The opposite of encryption is called decryption or decryption. Encryption is a method of converting the original data, called plain text or clear text or ordinary text, into a form that appears random and unreadable, called cipher text. Plain text is either in a form that can be understood by a human or a computer (executable code). Once it is converted into cipher text, neither human nor machine can process it properly unless it is decrypted.

Thus cryptography is a method of storing and transmitting data in a form that can only be read and processed by the person who organizes it. It is the science of protecting information by encoding it in an unreadable format. Cryptography is an effective way of protecting sensitive information as it is stored on a medium or transmitted through a network communication path. Applications of cryptography include ATM cards, computer passwords and electronic commerce. Cryptography has now entered the product markets on a large scale and every citizen of developed countries uses it on a daily basis. It is used for authentication and encryption (bank cards, wireless telephones, e-commerce, pay-TV), access control (car lock systems, ski lifts), payments (prepaid telephone cards, e-cash). It can be even more useful as an instrument of democracy with e-voting systems.



Cryptanalysis:

Cryptanalysis is the study of ways to obtain the meaning of encrypted information without having access to the required secret information. Generally, this involves knowing how the system works and also finding a secret key. Cryptanalysis is also known as code-breaking or code cracking. The cipher-text is usually the easiest part of a cryptosystem to obtain and is therefore an important part of cryptanalysis. The decryption and analysis of a code, cipher or encrypted text is called cryptanalysis. Cryptanalysis uses mathematical formulas to discover algorithmic vulnerabilities and to break into cryptography or information security systems.

Cryptanalysis (from Greek kryptos (underground), hidden (underground), and analinein (to understand)) is the art and science of analyzing information systems to study the hidden aspects of the system. Cryptographic techniques are used to subvert cryptographic security systems and to gain access to the contents of encrypted messages, even when the cryptographic key is unknown. Cryptanalysis is essentially an attempt to unencrypt encrypted data without using the key. The flip side of cryptography, cryptanalysis, is used to break codes by finding vulnerabilities. Besides being used by malicious hackers, cryptanalysis is often used by the military.

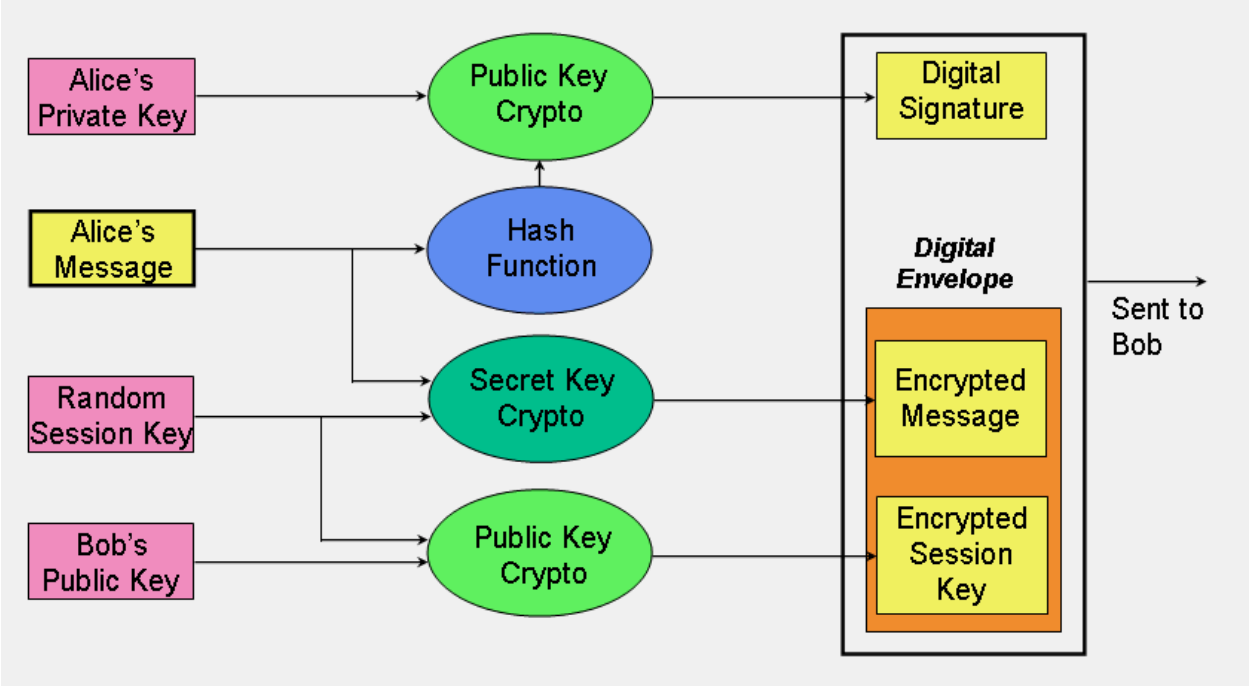
Cryptology:

Cryptology is the science of coding and decoding secret messages.
It is divided into cryptography, which deals with designing cryptosystems, and cryptanalysis, which deals with breaking cryptosystems.

Cryptology, the study of coded messages, can be traced back to Egypt around 1900 BC, when a scribe carved some hieroglyphic symbols into a rock in the tomb of Khnumhotep II.

Cryptology was not difficult then, as most people were illiterate and only the elite could read any written language. Pharaohs and potentates, kings and queens, presidents and dictators, and military commanders have used cryptology to hide their communications from their enemies ever since.

The word cryptology has its origins in ancient Greek. It is composed of two components, kryptos, meaning hidden, and logos, meaning word. Cryptology is thought to be as old as writing, and sign language can be considered the oldest form of cryptology. From the beginning of civilization to the advent of large-scale electronic communications, cryptology was primarily used to protect military and diplomatic communications. Since then it has become a part of common man's communications as well, as more and more people are turning to electronic communications such as e-mail. For example, in the past, cryptology was used by the military during war to ensure that strategies were not revealed to opponents. Cryptology is the science concerned with keeping data secure in communications and storage, usually in a secret form. It includes both cryptography and cryptanalysis. Security comes from legitimate users being able to convert information based on a secret key or keys. The resulting cipher, however, is generally unintelligible and undetectable without the secret key. It can be decrypted by anyone who knows the key to recover the hidden information or authenticate the source. Secrecy, though still an important function in cryptology, is often not the primary purpose of using transformations, and the resulting transformation may be considered simply a cipher.



Steganography

Steganography is similar to cryptography but adds another dimension to it. In this method, individuals not only want to protect the confidentiality of information by concealing it, but they also want to ensure that no unauthorized person can confirm that any information is even available. For example, invisible watermarking. In steganography, an uninvited person is unaware of the fact what information is hidden in the observed data, while in cryptography, the uninvited person is usually aware of the data being transmitted because they can see the coded message. Steganography is the practice of hiding a file, message, image, or video within another file, message, image, or video. The word steganography combines the Greek words steganos (στεγανός) meaning 'protected' and graphein (γράυειν) meaning 'writing'. The first documented uses of steganography can be traced back to 440 BC when Herodotus mentioned two examples in his Histories. Histiaeus had his most trusted servant shave his head and mark a message and send it to his vassal, Aristagoras. Additionally, Demetris sent a

warning about an impending attack on Greece by writing a message directly on the wood of a wax tablet before applying beeswax to its surface.

Another documented use of steganography was by Johannes Trithemius in 1499 in his Steganographia (Secret Writing), a treatise on cryptography and steganography disguised as a book on magic. Typically, hidden messages appear to be on (or part of) something else: images, writing, or some other protected text.

For example, the hidden message might be in invisible ink between the visible pages of a private letter. Steganographic schemes follow Kerckhoff's principle. The advantage of steganography over cryptography is that the intended secret message does not itself draw attention as an object of investigation. Cryptography is the practice of protecting the content of a message, while steganography sends a secret message while concealing the content of the message.

Steganography also involves hiding information within computer files.

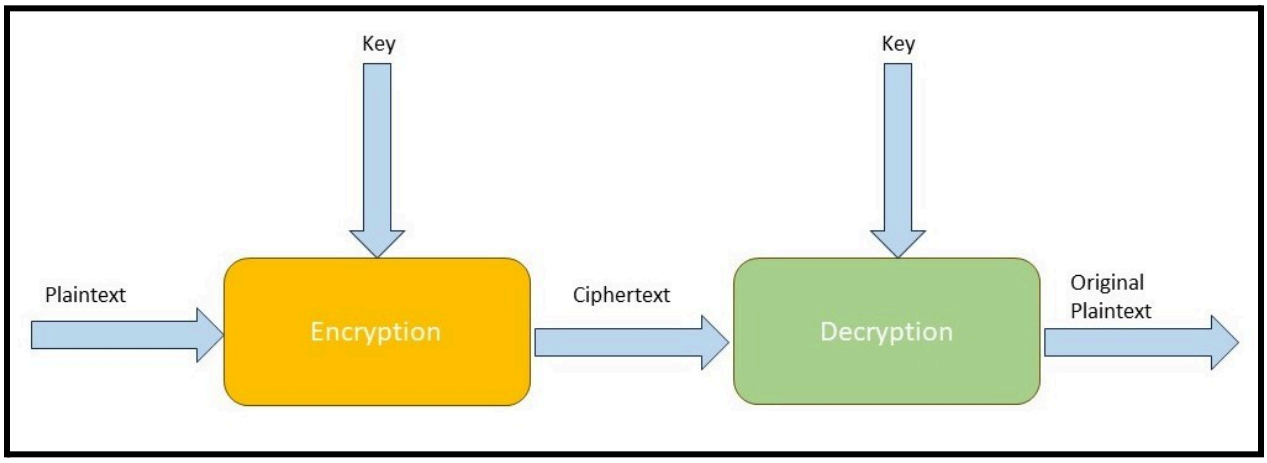
Digital steganography may involve steganographic coding within a transport layer in an electronic communication, such as a document file, image file, program or protocol. Media files are ideal for steganographic communication because of their large size.

Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, HTML or floppy disks) with bits of discrete, invisible information. This hidden information may be plain text, cipher text or even images. Sometimes steganography is used when encryption is not allowed, or more commonly, steganography is used to complement encryption. An encrypted file can never hide information using steganography, so even if the encrypted file is deciphered, the hidden message is not seen.

The difference between steganography and cryptography is that in cryptography, one can tell that a message is encrypted, but one cannot decode the message without knowing the proper key. In steganography, it may not be difficult to decode the message, but most people cannot detect the presence of the message.

History of Cryptography - The art of cryptography can be traced back to the art of writing. As civilizations developed, humans were organized into tribes, groups and empires. This led to the emergence of ideas such as warfare, supremacy and politics. These ideas fueled the natural need of people to communicate secretly with select recipients, which also ensured the continued development of cryptography. Cryptography has its roots in Roman and Egyptian civilizations. Hieroglyphs - The Earliest Cryptographic Technique: The earliest known cryptographic symbols can be traced back to the use of ‘hieroglyphs’. About 4000 years ago, Egyptians communicated using messages written in hieroglyphs. This code was known only to scribes who transmitted messages on behalf of kings.

Later, scholars started using simple mono-alphabetic substitution ciphers during 500 to 600 BC. This involved replacing letters of the message with other letters with certain secret rules. These rules became the key to recovering the original message from a distorted message. The first Roman method of cryptography, known as the Caesar shift cipher, relied on shifting the letters of the original message by an agreed number, and then the receiver of the message would shift the letters of the distorted message backwards by the same number to obtain the original message.



Development of Cryptography - During the European Renaissance and after, various Italian and Papal states rapidly spread cryptographic techniques. Various analysis techniques were researched in this era to break the secret codes. Improved coding techniques came into existence in the 15th century like Wigner coding, which offered to carry messages with multiple characters in the message instead of just a single number. After the 19th century, cryptography evolved from encryption to a more sophisticated art and science of information security. In the early 20th century, the invention of

mechanical and electromechanical machines such as the Enigma rotor machine provided more advanced and efficient means of coding information.

During the period of World War II, both cryptography and cryptanalysis became mathematical.

With the advancements in this field, government organizations, military units and some business houses started adopting the applications of cryptography. They used cryptography to protect their secrets from others. Currently, the advent of computers and the Internet has made effective cryptography accessible to the common people.

Medieval Cryptography: Around 1000 CE, inspiration from religion to analyze the Quran led to the invention of frequency analysis techniques for breaking monoalphabetic substitution ciphers. This was the most fundamental cryptanalytic advance until World War II. All ciphers remained vulnerable to this cryptanalytic technique until the invention of polyalphabetic ciphers by Alberti in 1465. Cryptanalysis makes use of the fact that the frequencies of letters in most English texts are not equal. For example, m occurs more often than c. Cryptography became important as a result of political competition and religious revolution. For example, during the Renaissance and later in Europe, the citizens of various Italian states, the Papal States, and the Roman Catholic Church were responsible for the rapid growth of cryptographic techniques, some of which predate Alberti's advances. The widespread use of cryptography for a variety of purposes during this period caused dismay to users.

Cryptanalysis and secret courier betrayal are featured in the Babington Plot, which led to the infighting of Mary, Queen of Scots, with the intent to assassinate Queen Elizabeth I. Cryptography and its misuse were also involved in the plot that led to the execution of Mata Hari and the tangle of Dreyfus' convictions and imprisonment in the early 20th century. In short, frequency of letter analysis cannot be applied to some ciphers. Lankeier, written by Ernest Vincent Wright, does not have a letter m. Even more impressive is the book Pencil by George Perec, written in French, which does not have a m. In these cases, frequency of letter analysis fails to unravel the secret message. In addition to the use of frequency of letter analysis, in this period, cryptography was used in both positive and negative ways. Some misuses ruined the users, while some used correctly brought innumerable benefits.

Cryptography from the 1800s to World War II:

Although cryptography has a long and complex history, it did not arise until the 19th century from much more than a discussion of encryption or cryptanalysis (the science of finding weaknesses in cryptosystems). Examples of the latter include Charles Babbage's Crimean War-era work on the mathematical cryptanalysis of polyalphabetic ciphers, which was later rediscovered and published by the Prussian Friedrich Kasiski.

Understanding cryptography at this time was generally a matter of strict rules of thumb. Later in the 19th century, see the cryptographic writings of Auguste Kerckhoff. Edgar Allan Poe used systematic methods to solve ciphers in the 1840s. In particular, he

advertised his abilities in the Philadelphia paper Alexander's Weekly (Express), inviting submissions for cipher solutions, of which he proceeded to solve nearly all. His success caused a public stir for some months. He later wrote an essay on methods of cryptography which proved useful as an introduction to the new British cryptanalysts attempting to break German codes and ciphers during World War I. In 1917, Gilbert Vernam proposed a teletype cipher in which a plaintext message is combined character by character to produce ciphertext, by placing a pre-prepared key on paper tape. This led to the development of one-time pads and the use of electronic devices as cipher machines.

Mathematical methods flourished in the period from 1932 to the year before World War II (notably in William F. Friedman's application of statistical techniques to cryptanalysis and cipher development and in Marian Rejewski's initial break of the German army's version of the Enigma system). Both cryptography and cryptanalysis became far more mathematical after World War II. Nevertheless, with the widespread availability of computers and the Internet as a means of communication, cryptography was increasingly being used by anyone other than national governments or similar large enterprises. During World War II, mechanical and electromechanical cryptographic cipher machines were widely used, although impractical hand mechanisms continued to be used. Developments were made in both practical and mathematical cryptography during this period. Germany made extensive use of an electromechanical rotor-based cipher system called Enigma in several forms. Marion Reze-Walker broke the early German military Enigma system in 1932 using theoretical mathematics. This was the

greatest breakthrough in cryptanalysis in thousands of years. After 1940, US Navy cryptographers in collaboration with British and Dutch cryptographers penetrated several Japanese Navy cryptosystems. The breaking of one of those codes, JN-25, famously led to the American victory at the Battle of Midway. An American army group, SIS, managed to break the highest security Japanese diplomatic cipher system, an electromechanical stepping switch machine called Purple by the Americans, even before World War II began. The Americans referred to secret information resulting from cryptanalysis, perhaps specifically from the Purple machine. The British eventually settled on ‘Ultra’ for intelligence resulting from cryptanalysis, especially from enciphered messages given by various Enigmas. The earlier British term for Ultra was ‘Boniface’. The German army also made several mechanical attempts at the one-time pad. Bleyle Park employed him to design the world's first programmable digital electronic computer, Colossus, to assist with the Fish cipher and Max Newman and his colleagues with cryptanalysis. The German Foreign Office began using one-time pads in 1919.

Modern Cryptography -

Modern cryptography is the cornerstone of computer and communication security. Its foundations are based on various concepts of mathematics such as number theory, computational-complexity theory, and probability theory.

The era of modern cryptography really began in 1949 with Claude Shannon, the father of mathematical cryptography. In 1976, two major public advances occurred. The first

was DES (Data Encryption Standard). The second development in 1976 was perhaps even more important, because it fundamentally showed that cryptosystems could work. It was the publication of the research paper in cryptography by Whitfield Diffie and Martin Hellman that gave a new direction to cryptography. This paper presented a new method of distributing cryptographic keys to solve one of the fundamental problems of cryptography, that is, key distribution, and it came to be known as Diffie-Hellman key exchange. In short, during this period, computers started to be used extensively for cryptographic purposes, for example, DES, 56-bit size encryption. Even though computer based encryption is considered modern and secure, it is still insufficient to protect the secret message from brute force attack. As a result, advanced cryptographic techniques are being developed like never before. Recently, quantum cryptography has emerged, which has high throughput as well as high accuracy. It can be predicted that quantum technologies will become the new era of truly unbreakable cryptography.

What is Cryptanalysis?

The art and science of breaking cipher text is called cryptanalysis. Cryptanalysis is a sister branch of cryptography and they both co-exist. Cryptographic process results in the decryption of cipher text for transmission or storage. It involves the study of cryptographic mechanisms in order to break them. Cryptanalysis is also used during the design of new cryptographic techniques to test their security strengths. Note:

Cryptography deals with the design of cryptosystems, while cryptanalysis studies the breaking of cryptosystems.

Security Services of Cryptography:

The primary purpose of using cryptography is to provide the following four fundamental information security services. We will look at the possible goals accomplished by cryptography.

1. Confidentiality:

Confidentiality is the fundamental security service provided by cryptography. It is a security service that keeps information away from unauthorized persons. It is sometimes referred to as confidentiality or secrecy. Confidentiality can be achieved through several means, starting from physical security to the use of mathematical algorithms for data encryption.

2. Data Integrity:

The security service that deals with identifying any alteration in the data is called data integrity. Data can be modified intentionally or accidentally by an unauthorized entity. Data integrity cannot prevent alteration of data but provides a means to detect if the data has been tampered with in an unauthorized manner.

3. Authentication:

Authentication provides the identity of the originator. It confirms to the recipient that the received data has been sent by an identified and verified sender only. There are two types of authentication service -

Message Authentication: Identifies the originator of the message without the router or system sending the message.

Entity Authentication: Assures that the data has been received from a particular entity, such as a particular website.

In addition to the originator, authentication can also provide assurance about other parameters related to the data such as date and time of creation/transmission.

Cryptosystem:

A cryptosystem is an implementation of cryptographic techniques and their accompanying infrastructure to provide information security services. A cryptosystem is also called a cipher system.

Power of a cryptosystem:

The power of an encryption method is the algorithm, the secrecy of the key, the length of the key, the initialization vector, and their working together. When power in encryption is discussed, it refers to how difficult it is to understand the algorithm or the key, which is not made public. Breaking a key is to process an infinite number of possible values in the hope of finding a value that can be used to decrypt a specific message. Strength is related to the key processing power and the amount of time it takes to break the key or figure out the key value. Breaking a key means trying every possible key value until the resulting plaintext is obtained. Depending on the algorithm and the length of the key, this can be a very easy task or a task close to impossible. If a key can be broken with a Pentium-2 processor in three hours, the cipher is not strong at all. If the key can only be broken using a thousand multiprocessing systems, and it takes 1.2 million years, it is very strong. The goal of designing an encryption method is to make it more time consuming. Another name for cryptography strength is the work factor, which is an estimate of the effort an attacker has to put into breaking an encryption method. The strength of the security mechanism should be used in correlation with the sensitivity of the encrypted data.

Even if the algorithm is very complex and deep, there are other problems within the encryption that may weaken the strength of the encryption methods. Since the key is usually the secret value needed to encrypt and decrypt messages, improper security of the key can weaken the encryption strength. An extremely strong algorithm can be used

by using a large keyboard, and a large and random key value, which is necessary for strong encryption, but if a user shares his key with others, the other pieces of the equation don't really matter.

Goal of a Cryptosystem:

The goal of cryptography is to make it possible to exchange a message between two persons without the other person understanding the message. There is no end to the methods that can be used to achieve this. Here we will be concerned with methods of transforming a text in such a way that the recipient can undo the transformation and discover the original text.

The original text is usually called the cleartext and the encoded or transformed text is called the ciphertext. The conversion from cleartext to ciphertext is called encoding or enciphering, and the opposite operation is called decoding or deciphering. If we are trying to read a secret message that was not meant for us and we do not know the encoding method initially, this process is called cracking the code.

Generally, the more ciphertext we have, the easier it is to break the code. So usually the coding mechanism must be changed regularly.

For example, if a coding scheme contains a keyword, and a different keyword is used each day, there may never be enough ciphertext to decode the message. But if we

change the encoding each day, we must have a secure method of getting the new keyword.

Cryptosystems can provide confidentiality, authenticity, integrity and non-repudiation services. It does not provide availability of data or system. Confidentiality means that unauthorized persons cannot access the information. Authenticity refers to verifying the source of a message so that the sender can be correctly identified. Integrity provides assurance that the message was not modified at the time of transmission, accidentally or intentionally. Non-repudiation means that a sender cannot deny sending a message at a later date, and a recipient cannot deny receiving it. Different types of messages and transactions require one or more services that can supply a high level of security. The military and intelligence departments are more concerned about keeping information confidential, so they will choose encryption mechanisms that provide a high level of confidentiality. Financial institutions care about privacy, but care more about the integrity of the data being transmitted, so they choose encryption mechanisms that provide a high level of confidentiality. This may be different from the military's encryption methods.

Components of a Cryptosystem:

The various components of a basic cryptosystem are as follows -

Plaintext: It is the data protected during communication.

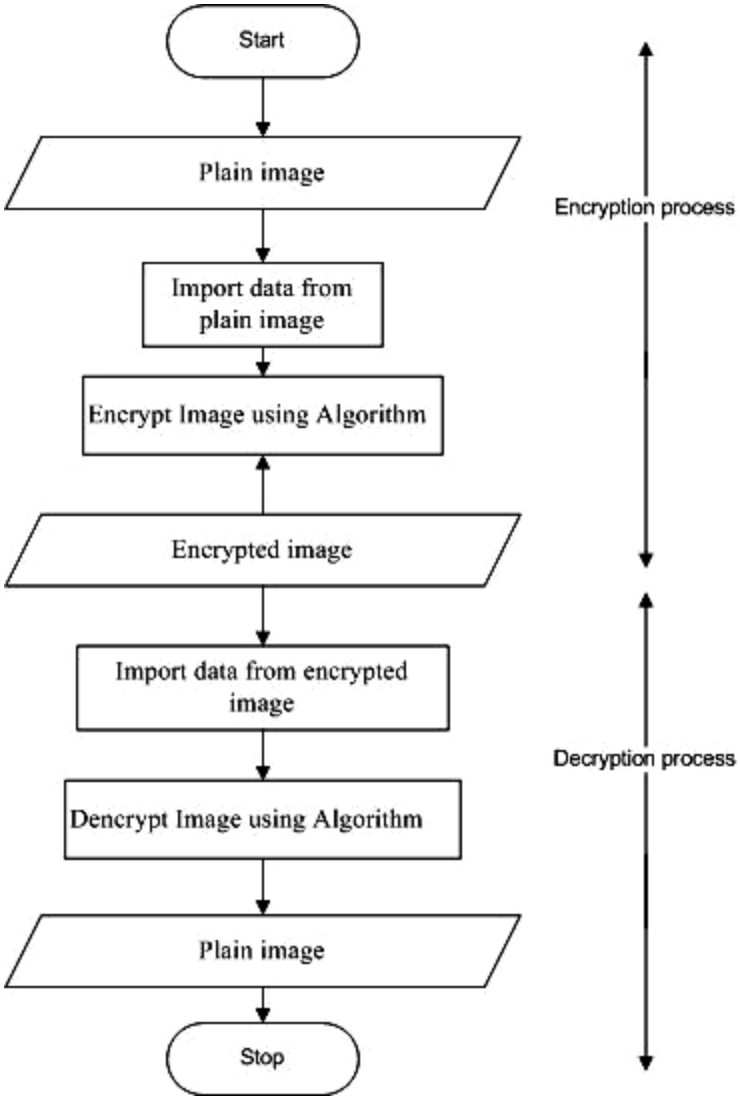
Encryption Algorithm: It is a mathematical procedure that generates a ciphertext for a given plaintext and encryption key. It is a cryptographic algorithm that takes plaintext

and encryption key as input and generates a ciphertext. Ciphertext: It is a coded version of plaintext generated by an encryption algorithm using a specific encryption key.

Ciphertext is not protected. It resides in the public domain. It can be intercepted by anyone who has access to the communication system.

Decryption algorithm: It is a mathematical procedure that generates a unique plaintext for any given ciphertext and decryption key. It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and gives a plaintext as output. The decryption algorithm essentially reverses the encryption algorithm and is thus closely related to it.

Encryption key: This is a value known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext to calculate the ciphertext.



Decryption key: This is a value known to the receiver. The decryption key is related to the encryption key, but it is not always the same. The recipient inputs the decryption key along with the ciphertext into the decryption algorithm to calculate the plaintext.

Types of Cryptosystems

Basically, there are two types of cryptosystems that help in encryption-decryption (encryption-unencryption) in the system -

1. Symmetric Key Encryption

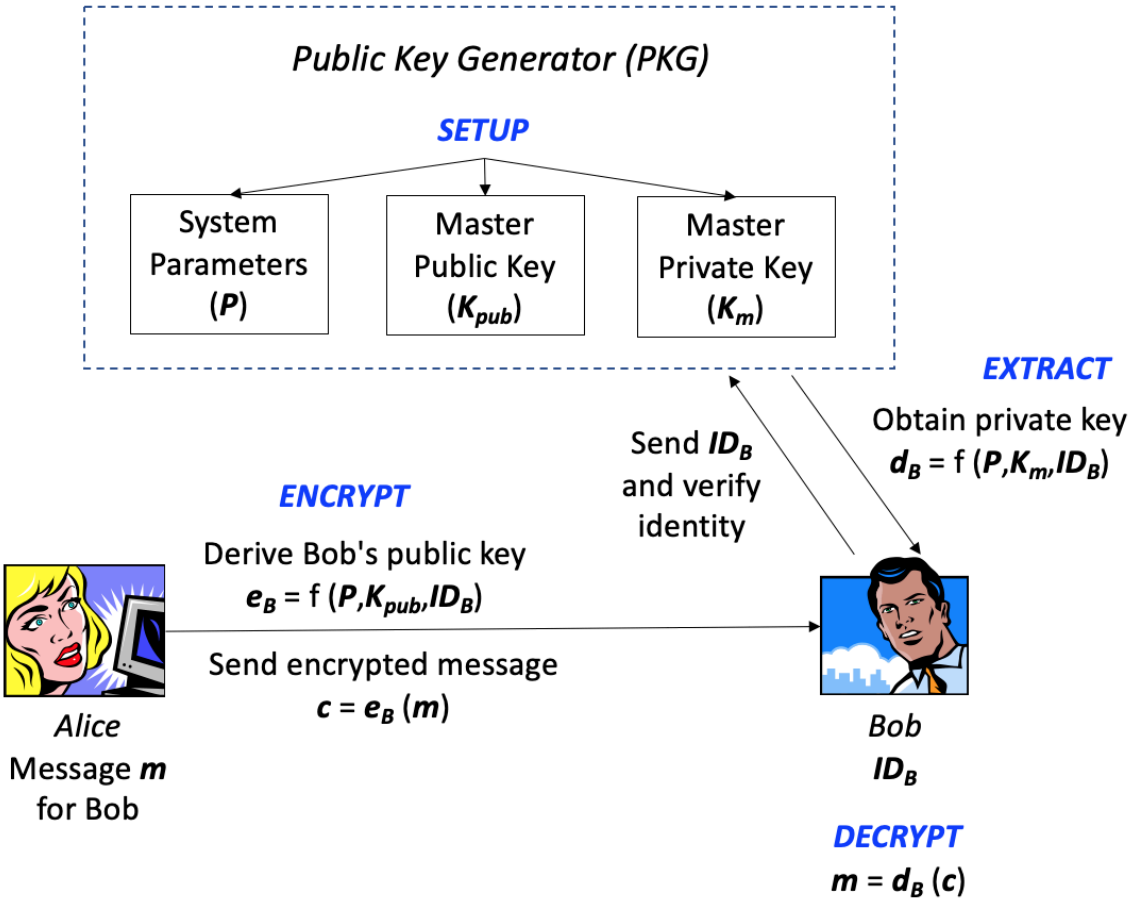
2. Asymmetric Key Encryption

The main difference between these cryptosystems is the relationship between the encryption and decryption keys. Logically, in any cryptosystem, the two keys are closely linked. It is practically impossible to decrypt the ciphertext with a key that is not related to the encryption key.

Challenge of Public Key Cryptosystems

There is a significant challenge in public key cryptosystems. A user has to trust that a public key that he is using in communication with another person is indeed the public key of that person and has not been spoofed by a malicious third party.

This usually involves a trusted third party through a public key infrastructure (PK). The third party securely manages and certifies the authenticity of the public keys. When a third party is requested to provide the public key for any correspondent person X, trust has to be placed to obtain the correct public key.



The third party satisfies itself about the user identity by authentication or some other process that X is unique or globally unique. The most common method for providing a verified public key is to provide a certificate that is digitally signed by a trusted third party.

Because of the advantages and disadvantages of both systems, symmetric key and public key cryptosystems are often used together in practical information security systems.

Public key cryptosystems have a significant challenge. The user has to trust that a public key he or she is using in communication with another person is actually that person's public key and has not been spoofed by a malicious third party.

This usually involves a trusted third party via a public key infrastructure (PKI). The third party securely manages and certifies the authenticity of the public key. When a third party is requested to provide the public key for any correspondent person X, trust is required to obtain the correct public key. The third party satisfies itself about the user identity by authentication or some other process that X is unique or globally unique. The most common method to provide a verified public key is to provide a certificate that is digitally signed by the trusted third party. Because of the advantages and disadvantages of both systems, symmetric key and public key cryptosystems are often used together in practical information security systems.

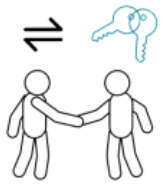
Cryptography Merits and Demerits

Nowadays, networks have become global and information has taken the digital form of bits and bytes. Critical information is now stored, processed and transmitted in digital form on computers and open communication systems.

Since information has a vital role, attackers are targeting computer systems and may steal sensitive information or disrupt critical information systems.

Comparison between
Symmetric and **Asymmetric** Key Cryptography

Symmetric Key Cryptography	Asymmetric Key Cryptography
Utilizes a singular key for both ciphering and deciphering	Employs a duo of keys for encryption and decryption (public and confidential keys)
Key magnitude is usually brief	Key magnitude is commonly lengthy
Speeds up encryption and decryption	Slows down encryption and decryption
Key administration is uncomplicated	Key administration is more intricate
More adaptable in extensive deployment situations	Less adaptable in extensive deployment situations
Optimal for use cases where data requires prompt encryption and decryption	Optimal for use cases where secure correspondence is necessary



Public Key Encrypts
Private Key Decrypts

Modern cryptography provides a solid set of techniques to ensure that the malicious purposes of the attacker are thwarted while ensuring legitimate users have access to information. Now we will discuss the limitations of cryptography as well as the benefits derived from the future of cryptography. Properties of Cryptography:

Cryptography is an essential information security tool. It provides four most basic services of information security:

Confidentiality - Encryption techniques can protect information and communication from unauthorized publication and access to information.

Authentication - Cryptographic techniques such as DTH and digital signatures can protect information against falsification.

Data Integrity - Cryptographic hash functions are playing an important role in assuring users about data integrity.

Non-repudiation - Digital signatures provide non-repudiation service to protect against dispute arising due to denial of the transmitted message by the sender. All these fundamental services provided by cryptography have enabled the use of computer systems to conduct business over networks in a highly efficient and effective manner.

Disadvantages of Cryptography:

In addition to the four fundamental elements of information security, there are other issues that affect the effective use of information:

Strongly encrypted, authenticated, and digitally signed information may be difficult to access even for a legitimate user at a critical time of decision making.

The network or computer system may be attacked and rendered non-functional by an attacker.

High quality, one of the fundamental aspects of information security, cannot be ensured through the use of cryptography. Other methods are needed to avoid threats such as denial of service or complete breakdown of the information system. Another fundamental requirement of information security of selective access control also cannot be achieved through the use of cryptography. It requires the use of administrative controls and procedures.

Applications of Cryptography:

Privacy:

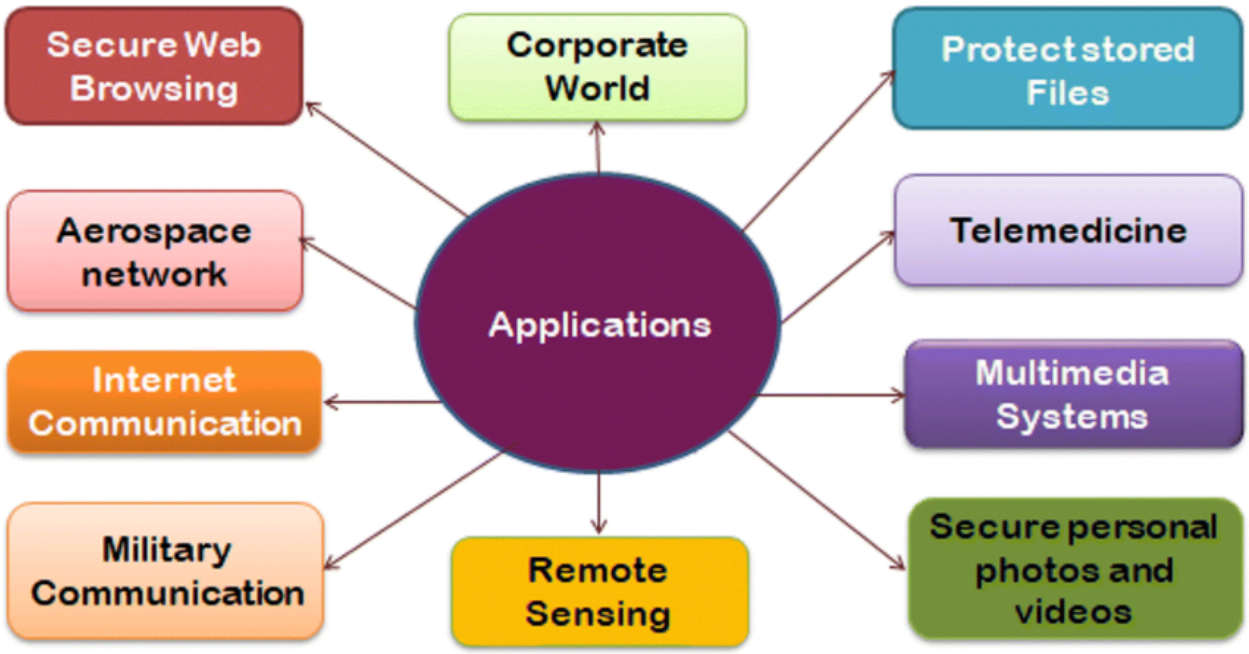
Privacy is the most obvious application of cryptography. Cryptography can be used to enforce privacy by encrypting information from being personal. For someone to read this personal data, it must first be decrypted.

Authentication:

Authentication is a process through which certain information can be proven and verified. Sometimes it is necessary to verify the origin of a document, the identity of the sender, the time and date the document was signed, the identity of the computer or user, and so on.

Digital Signature:

Digital signatures are a cryptographic means by which many of these can be verified. A digital signature of a document is a piece of information based on both the document and the signer's personal key.

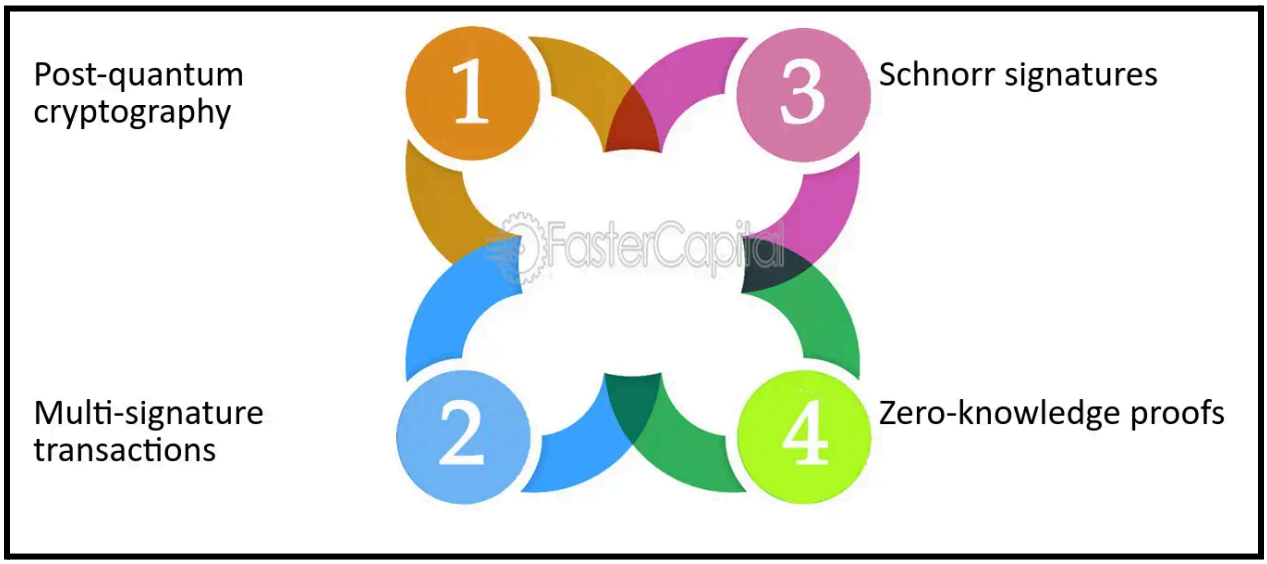


Both digital signatures and handwritten signatures rely on the fact that it is very difficult to find two people with the same signature. Public key cryptography is used to calculate digital signatures by associating something unique with each person. When public-key cryptography is used to encrypt a message, the sender encrypts the message with the public key of the intended recipient. When public key cryptography is used to calculate a digital signature, the sender encrypts the digital fingerprint of the document with his or her personal key. Anyone with access to the signer's public key can verify the signature.

Digital Envelope:

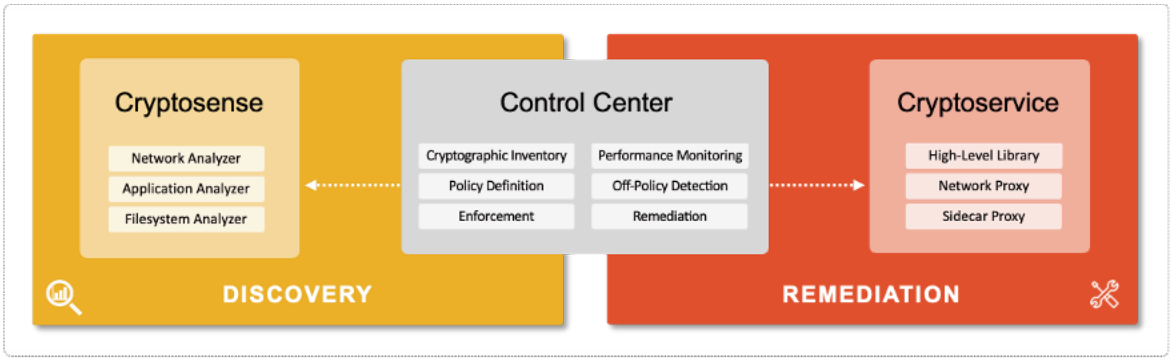
The digital envelope holds the encrypted message using public-key cryptography and the encrypted secret key. While digital envelopes typically use public-key cryptography to encrypt the secret key, it is not necessary.

Future of Cryptography: Elliptic curve cryptography (EC) has already been invented but its advantages and disadvantages are not yet fully understood. EC allows encryption and decryption to be done in a very short time. Thus it allows passing of high volumes of data with the same security. However, as other methods of encryption, EC must also be tested and proven before it can be accepted for government, commercial and personal use.



Quantum computing is a new phenomenon. Modern computers store data using a binary format called a ‘bit’ which can store 1 or 0, while a quantum computer stores data using quantum superposition of more than one state. These multi-valued states are

stored in quantum bits or qubits. This allows calculation of numbers at a much faster rate than conventional transistor processors.



To understand the power of quantum computers, consider RSA-640 (RSA-640), a number with 193 digits, which can be factored by eighty 2.2Gb computers in a period of 5 months, while a quantum computer can do the same task in less than 17 seconds.

With these facts in mind, modern cryptography has to look for harder problems to compute or devise entirely new techniques to store the goals currently offered by modern cryptography.

Chapter 2

Ancient Encryption

Ancient cryptography is the best starting point to learn cryptosystems without prior knowledge of cryptography and high-level mathematics. Here we will review ancient cryptosystems. Ancient cryptosystems refer to those that are pencil-and-paper based and have been in use until the mid-twentieth century. These include shift, affine, simple substitution, transposition, Hill, and Vigenere ciphers. Ancient encryption is also called symmetric key encryption where the key used for encryption and decryption is the same.

Caesar or Displacement Cipher:

In the first century BCE, Julius Caesar developed the displacement cipher. Based on the displacement key, each letter in the alphabet is replaced by a new shifted letter. The message which is encrypted by substituting it by the corresponding new letter is called the plain text, and the new message produced by the substitution is called the cipher text. The key describes the number of letters with the help of which the letters of the plain text are replaced in the cipher text. It is also called mono-alphabetic cipher in which each letter of the plain text is replaced by another letter to produce the cipher text. It is a simple form of substitution cipher scheme. This cryptosystem is commonly

known as displacement cipher. The idea is to replace each alphabet with another alphabet shifted by some fixed number between 0 and 25.

For this type of scheme, both the sender and the receiver agree on a secret displacement number to shift the alphabet by. This number, which lies between 0 and 25, becomes the encryption key. The name ‘Caesar cipher’ is sometimes used to describe a shift cipher.

Decryption:

Each character of the ciphertext is converted into a plaintext character as follows:

N Shift(5) = I

F Shift(5) = A

R Shift(5) = M

F Shift(5) = A

X Shift(5) = S

Y Shift(5) = T

Z Shift(5) = U

I Shift(5) = D

J Shift(5) = E

S Shift(5) = N

Y Shift(5) = T

Simple Substitution Cipher:

A monoalphabetic or simple substitution cipher is one in which the ciphertext alphabet is a rearrangement of the plaintext alphabet. Substitution ciphers have 26! possible permutations, which are actually very insecure and can be easily solved using the frequency of letters. Simple substitution is a perfect method for understanding messages that can be easily decrypted by visual understanding.

It is an advanced form of Caesar cipher. Instead of shifting the letters by some numbers, this scheme uses some permutations of letters in the alphabet. For example, 133.1 and 133.2 are two obvious permutations of all letters in the alphabet. A permutation is nothing but a connected group of the alphabet. With 26 letters in the alphabet, the possible permutations are 26!, which is equal to 41026. The sender and receiver can choose any of these possible permutations as the ciphertext alphabet.

Monoalphabetic and Polyalphabetic Ciphers:

A monoalphabetic cipher is a substitution cipher in which, for a given key, the cipher alphabet for each base alphabet is fixed in the encryption process. For example, if l is encrypted as l, then for any number that appears in that plaintext, l will always be encrypted as l.

The substitution ciphers discussed above are monoalphabetic. These ciphers are susceptible to cryptanalysis.

A polyalphabetic cipher is a substitution cipher in which the cipher alphabet may differ from the original alphabet at different places during the encryption process. The next two ciphers, Playfair and Vigenere, are polyalphabetic ciphers.

A cipher is polyalphabetic if a given letter of the alphabet will not always be encrypted by the same ciphertext letter, and as a result, cannot be described by a set of ciphertext alphabets corresponding to a set of plaintext alphabets.

The simplest method of generating a polyalphabetic cipher is to combine different monoalphabetic ciphers.

A problem with monoalphabetic ciphers is that letters occur with some frequency in a language. This frequency can be worked out for the plaintext letters and the ciphertext letters of the predicted message, and, after some analysis, the cipher is relatively easily cracked.

Homonymous ciphers change the situation by making the frequencies of the ciphertext letters equal. It uses the following rules:

For unique decryption, the groups corresponding to specific plaintext letters must be distinct. The number of ciphertext symbols allocated to the plaintext is determined by the frequency of that letter.

Playfair Cipher:

In this method, pairs of letters are encrypted instead of single letters as compared to simple substitution ciphers.

In Playfair Cipher, a key table is created initially. The key table is a 5 × 5 grid of the alphabet which serves as the key to encrypt the plaintext. Each of the 25 letters must be unique and one letter of the alphabet (usually Sh) is omitted from the table as we need only 25 letters of the alphabet instead of 26. If the plain text contains sh, it is replaced by p.

For example if the sender and receiver agree on a particular key, say 2ञ्चवत्प।स". Then in the key table, the first letters of the phrase ञ्चवत्प।स (excluding duplicate letters) will be placed going from left to right. The rest of the table will be filled with the remaining letters of the alphabet in natural order.

Vigenere Cipher:

The Vigenere cipher is an example of a polyalphabetic substitution cipher. A polyalphabetic substitution cipher is similar to a monoalphabetic substitution cipher,

with the difference being that the cipher alphabet changes periodically while the message is being deciphered.

This makes the cipher reducible to cryptanalysis using letter frequencies.

It was developed by Blaise de Vigenere and came to be known as the Vigenere cipher in 1585. He used a table known as the Vigenere square to decipher messages. Here we will discuss two different methods of Vigenere cipher namely autokey method and keyword method.

Encryption:

To decipher a message using Vigenere autokey method, the sender and receiver must first agree on a primary key. The primary key is a single letter that will be added to the beginning of the message to form the key. The sender will encrypt the message by writing plain text on one line and the key on the line below it. The sender will use the plain text and the key letters to select one row and one column in the Vigenere square. The selected line is the line that has the plaintext letter j in the first column and the selected column is the column that has the key letter j in the first row. A ciphertext letter is the letter that appears at the position corresponding to the selected row and column in the Vigenere square.

In the following example, to find the ciphertext letter, first locate the row in the Vigenere square that corresponds to the plaintext letter j.

Then locate the column corresponding to the key letter s. The letter at which the row and column intersect will be the ciphertext letter, here the ciphertext letter is m. Continuing in this sequence, obtain the ciphertext letter for each pair of letters.

primary key

plain-text T O B E O R N O T T O B E

Key S L T O B E O R N O T T O B

Cipher-text H P F S F E B H M H P F

Computer programs for various methods:

1. Encryption

```
#include<stdio.h>
int main()
{
char message[100], ch;
int i, key;
printf("Enter a message to encrypt: ");
gets(message);
printf("Enter key: ");
scanf("%d", &key);
for(i = 0; message[i] != '\0'; ++i){
ch = message[i];
if(ch >= 'a' && ch <= 'z'){
```

```
ch = ch + key;
if(ch > 'z'){
ch = ch - 'z' + 'a' - 1;
}
message[i] = ch;
}
else if(ch >= 'A' && ch <= 'Z'){
ch = ch + key;
if(ch > 'Z')
{
ch = ch - 'Z' + 'A' - 1;
}
message[i] = ch;
}
}
printf("Encrypted message: %s", message);
return 0;
}
```

Output

```
Enter a message to encrypt: axzd
Enter key: 4
Encrypted message: ebdh
```

Decryption

```
#include<stdio.h>
int main()
{
char message[100], ch;
int i, key;
printf("Enter a message to decrypt: ");
gets(message);
printf("Enter key: ");
scanf("%d", &key);
for(i = 0; message[i] != '\0'; ++i){
ch = message[i];
if(ch >= 'a' && ch <= 'z'){
ch = ch - key;
if(ch < 'a'){
ch = ch + 'z' - 'a' + 1;
}
message[i] = ch;
}
else if(ch >= 'A' && ch <= 'Z'){
ch = ch - key;
if(ch < 'A'){
ch = ch + 'Z' - 'A' + 1;
```

```
}
message[i] = ch;
}
}
printf("Decrypted message: %s", message);
return 0;
}
```

Output

```
Enter a message to decrypt: ebdh
Enter key: 4
Decrypted message: axzd
```

2.
```
#include<stdio.h>
int main()
{
char *message,*emessage,*dmessage;
int i,j=0,k,key,temp;
clrscr();
printf("\nEnter the key\n");
scanf("%d",&key);
key=key%26;
printf("\nEnter message\n");
```

```
fflush(stdin);
gets(message);
for(i=0;message[i]!=NULL;i++)
message[i]=tolower(message[i]);
for(i=0;message[i]!=NULL;i++)
{
//printf("%c ",message[i]);
if(message[i]==' ')
emessage[j++]=message[i];
else
{
if(message[i]>=48 && message[i]<=57)
{
temp=message[i]+key;
if(temp>57)
emessage[j++]=48+(temp-58);
else
emessage[j++]=temp;
}
else
{
if(message[i]>=97 && message[i]<=123)
{
```

```
temp=message[i]+key;
if(temp>122)
emessage[j++]=97+(temp-123);
else
emessage[j++]=temp;
}
else
emessage[j++]=message[i];
}
// printf("%c ",emessage[j]);
}
}
emessage[j]='\0';
printf("\n\n\nEncrypted message is\n\n");
for(i=0;emessage[i]!=NULL;i++)
printf("%c",emessage[i]);
// printf("\n end");
for(i=0,j=0;emessage[i]!=NULL;i++)
{
if(emessage[i]==' ')
dmessage[j++]=emessage[i];
else
{
```

```
if(emessage[i]>=48 && emessage[i]<=57)
{
temp=emessage[i]-key;
if(temp<48)
dmessage[j++]=58-(48-temp);
else
dmessage[j++]=temp;
}
else
{
if(emessage[i]>=97 && emessage[i]<=123)
{
temp=emessage[i]-key;
if(temp<97)
dmessage[j++]=123-(97-temp);
else
dmessage[j++]=temp;
}
else
dmessage[j++]=emessage[i];
}
}
}
```

```
dmessage[j]='\0';
printf("\n\n\nRetrieved message is\n\n");
for(i=0;dmessage[i]!=NULL;i++)
printf("%c",dmessage[i]);
getch();
return(0);
}
```

Output:

```
Enter the Key
3
Enter Message
Hello
Encrypted Message is
khoor
Retrieved message is
Hello
```

3. Encryption

```
#include<stdio.h>
int main(){
char arr[5][5]={"MONAR","CHYBD","EFGIK","LPQST","UVWXZ"};
char pt[10];
```

```
int i, j, r1=0, r2=0, c1=0, c2=0;
printf("Playfair Keymatrix\n==================\n");
for(i=0; i<5; i++)
{
for(j=0; j<5; j++)
printf("%c ", arr[i][j]);
printf("\n");
}
printf("Enter your plain text:");
scanf("%s",pt);
printf("Your plain text = %s", pt);
for(i=0; i<5; i++)
{
for(j=0; j<5; j++)
{
if(arr[i][j] == pt[0])
{
r1=i; c1=j;
}
if(arr[i][j] == pt[1])
{
r2=i; c2=j;
}
```

```
}
}
if(r1==r2) //when both characters in same row
{
if(c2==4) //for char in last column
printf("Ciphertext = %c%c \n", arr[r1][c1+1], arr[r2][0]);
else
printf("Ciphertext = %c%c \n", arr[r1][c1+1], arr[r2][c2+1]);
}
if(c1==c2)//when both characters in same column
{
if(r2==4) //for char in last row
printf("Ciphertext = %c%c \n", arr[r1+1][c1], arr[0][c2]);
else
printf("Ciphertext = %c%c \n", arr[r1+1][c1], arr[r2+1][c2]);
}
//when characters are not in a same row and column

if(r1 != r2 && c1 != c2)
{
printf("\nCiphertext = %c%c \n", arr[r1][c2], arr[r2][c1]);
}
return 0;
```

}

Output:

Playfair Keymatrix

==========

M O N A R

C H Y B D

E F G I K

L P Q S T

U V W X Z

Enter your plain text:IN

Your plain text = IN

Ciphertext = GA

Decryption

```
#include<stdio.h>
int main()
{
char arr[5][5]={"MONAR","CHYBD","EFGIK","LPQST","UVWXZ"};
char ct[10];
int i, j, r1=0, r2=0, c1=0, c2=0;
printf("Plaifair Keymatrix\n==========\n");
for(i=0; i<5; i++)
{
for(j=0; j<5; j++)
```

```
printf("%c ", arr[i][j]);
printf("\n");
}
printf("Enter your cipher text:");
scanf("%s",ct);
printf("Your cipher text is %s\n", ct);
for(i=0; i<5; i++)
{
for(j=0; j<5; j++)
{
if(arr[i][j] == ct[0])
{
r1=i; c1=j;
}
if(arr[i][j] == ct[1])
{
r2=i; c2=j;
}
}
}
if(r1==r2) //Rule2-when both characters in same row
{
if(c2==0) //for char in last column
```

```
printf("Plaintext = %c%c \n", arr[r1][c1-1], arr[r2][4]);
else
printf("Plaintext = %c%c \n", arr[r1][c1-1], arr[r2][c2-1]);
}
if(c1==c2)//Rule3- when both characters in same column
{
if(r2==0) //for char in last row
printf("Plaintext = %c%c \n", arr[r1-1][c1], arr[4][c2]);
else
printf("Plaintext = %c%c \n", arr[r1-1][c1], arr[r2-1][c2]);
}
//Rule4 when characters are not in a same row and column
if(r1 != r2 && c1 != c2)
{
printf("Plaintext = %c%c \n", arr[r1][c2], arr[r2][c1]);
}
return 0;
}
```

Output:

```
Plaifair Keymatrix
==================
M O N A R
C H Y B D
```

E F G I K

L P Q S T

U V W X Z

Enter your cipher text:NA

Your cipher text is NA

Plaintext = ON

Chapter 3

Public Key Encryption

Public Key Cryptography:

Public key cryptography, or asymmetric cryptography, is a cryptographic system that uses a pair of keys: a public key that can be widely disseminated, and a private key that is known only to the owner. The generation of such keys relies on a cryptographic algorithm based on mathematical problems to produce one-way functions. Effective security requires only private keys; public keys can be openly distributed without compromising security. In such a system, anyone can encrypt a message using the recipient's public key, but that encrypted message can only be decrypted with the recipient's private key.

Strong authentication is also possible. A sender can associate a message with a personal key to create a short digital signature on the message. Anyone with the corresponding public key can associate a message, affix a short digital signature on it, and a known public key to verify whether the signature was valid, i.e. was created by the owner of the corresponding personal key.

Public key algorithms are fundamental security elements in modern cryptosystems, applications, and protocols that assure confidentiality, authenticity, and non-repudiation of electronic communications and data storage. They are maintained by various Internet standards, such as Transport Layer Security (TLS), IPv6, and Gigabit Ethernet. Some public key algorithms provide key distribution and confidentiality (e.g., Diffie-Hellman key exchange), some provide digital signatures (e.g., Digital Signature Algorithm), and some provide both (e.g., RFID). Before the mid-1970s, all cipher systems were using symmetric key algorithms, in which the same cryptographic key is used by both the sender and the recipient, along with the underlying algorithm, which must be kept secret by both. By necessity, in every such system the keys had to be exchanged between the communicating parties in some secure way before any system could be used - a secure system. This requirement is never trivial and very quickly becomes unmanageable as the number of participants grows, or when secure systems for key exchange are not available, or when (as is sensible cryptographic practice), the keys change frequently. In particular, if the messages are secure from other users, a different key is required for each possible pair of users. In contrast, in a public key system, the public keys can be widely and openly disseminated - and only the individual key needs to be kept secure by its owner. An approach called asymmetric cryptography evolved to address the security issues posed by symmetric cryptography. This method solves the problem of secret key cryptography by using two keys instead of a single key. Asymmetric cryptography uses a pair of keys. In this process, one key is used for encryption, and the other key is used for decryption.

This process is known as asymmetric cryptography as both the keys are required to complete the process. These two keys are collectively known as the key pair. In asymmetric cryptography, one of the keys is independently distributable. This key is called the public key and is used for encryption. Hence, this method of encryption is also called public key encryption. The other key is the secret or personal key and is used for decryption. The personal key is not distributable. This key, as its name suggests, is personal to each communicating entity. In public key cryptography, data encrypted with a public key can only be decrypted with the same private key. Conversely, data encrypted with a private key can only be decrypted with the corresponding public key. Because of this asymmetry, public key cryptography is known as asymmetric cryptography. How does public key cryptography work? Let us see how it works in practice. Consider an example where a sender wants to send an encrypted file to a receiver. In this case, the receiver will receive a pair of keys, retain the private key and distribute the public key. Therefore, the sender has a copy of the public key of the receiver. The sender then encrypts the file using the recipient's public key and sends the encrypted file to the recipient. Since the key pairs are complementary, only the recipient's private key can decrypt this file. If someone else intercepts the file, they will not be able to decrypt the file, as only the recipient's private key can be used for decryption.

Public Key Encryption:

This method very clearly indicates that the data we send to the user can only be encrypted by the public key. Similarly, decryption can only be done by the personal key, which is supplied by the data recipient.

Therefore, there is very little chance of the data in transit being accessed or tampered with by someone else.

Hence, the messages can be exchanged securely. The sender and receiver do not need to share a key, as is required for symmetric encryption. All communications involve only the public key, and no personal key is ever transmitted or shared.

The above mechanism also brings forth that each recipient will have a unique key which he will use to decrypt the data encrypted by his counterpart's public key. Diffie and Hellman first discussed the process of asymmetric cryptography. The most common implementation of this process is the PSP algorithm.

Unlike symmetric key cryptography, we do not find the historical use of public-key cryptography. It is a relatively new concept.

Symmetric cryptography was suitable for organizations such as governments, military, and large financial corporations that were involved in classified communications.

With the proliferation of more insecure computer networks in the last few decades, a real need was felt to use cryptography on a large scale. Symmetric keys were found to be non-practical due to the challenges faced for key management.

This gave rise to public key cryptosystems. The most important properties of a public key encryption scheme are as follows:

Different keys are used for encryption and decryption. This is a property that distinguishes this scheme from symmetric encryption schemes.

Each recipient has a unique decryption key, usually called his/her personal key.
The recipient is required to publish an encryption key, called his/her public key.

Some assurance of the authenticity of a public key is required in this scheme to avoid spoofing by an adversary masquerading as the recipient. Generally, this type of cryptosystem involves a trusted third party that certifies that a particular public key belongs only to a specific person or entity. The encryption algorithm is complex, which prevents an attacker from deducing the ciphertext and encryption key from the plaintext.

Although the private and public keys are mathematically related, it is not possible to calculate the private key from the public key. In fact, the smart part of any public-key cryptosystem is to create a relationship between the two keys.

Advantages of Public Key Cryptography:

Public key encryption, in which a message is encrypted with the recipient's public key. This message cannot be decrypted by anyone who does not have the matching private key, who is thus considered the owner of that key and the person associated with the public key. It is used in an attempt to ensure confidentiality.

Digital signatures, in which a message is signed with the sender's personal key and can be verified by anyone who has the sender's public key. This verifies that the sender had access to the personal key, and is therefore likely to be the person associated with the public key. This also ensures that the message has not been tampered with, as the signature is mathematically bound to the message it was originally created with, and the verification will fail for practically any other message, even if it is identical to the original message.

The data security provided by public key cryptography is its main advantage. Public key cryptography remains the most secure protocol because users are never required to transmit or reveal their personal keys to anyone, making it less likely for cybercriminals to discover an individual's secret key at the time of transmission.

Public key cryptography also provides digital signatures that cannot be recreated. Public key cryptography requires each user to be responsible for protecting his or her own personal key, while personal key systems require users to share a secret key and

perhaps even trust a third party for transmission. With secret key systems, it is possible for senders to claim that the shared secret key was compromised by one of the parties involved in the process.

The most obvious application of a public key encryption system is in encrypting communications to provide confidentiality – a message that the sender encrypts using the recipient's public key can only be decrypted by the recipient's paired personal key. Since asymmetric key algorithms are almost always much more computationally intensive than symmetric ones, in many cases it is common to exchange a key using a key-exchange algorithm, then transmit data using that key and a symmetric key algorithm.

Challenges of Public Key Cryptography:

Speed is often cited as the most common challenge associated with public key cryptography.

Many private key cryptography methods are much more intensive than currently available public key encryption methods. One method of overcoming this challenge with public key cryptography is to combine it with private key systems to provide the security benefits of a public key system and the speed of a private (private) key system.

Another challenge associated with public key cryptography is that it is susceptible to attacks through spoofs and compromised certification authorities. When these attacks

occur, cybercriminals can impersonate almost anyone by choosing a public key certificate from a compromised authority. This allows cybercriminals to associate one public key with another user's name.

There are three types of public key encryption schemes. They are as follows, we will discuss them in the next sections:

1. RSA Cryptosystem
2. ElGamal Cryptosystem
3. Elliptic Curve Cryptography

RSA Cryptosystem:

This cryptosystem is an early system. It is still the most widely used cryptosystem today. This system was invented by three scholars Ron Rivest, Adi Shamir and Len Adleman and hence, it is called RSA Cryptosystem.

We will look at two aspects of RSA Cryptosystem, generation of key pairs and encryption-decryption algorithm.

RSA is one of the first public-key cryptosystems and is widely used for secure data transmission. In such cryptosystems the encryption key is public and is different from the decryption key which is kept secret (private). In Cryptosystems, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factorization problem. A Cryptosystems user creates and then publishes a public key

based on the two large prime numbers along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, and if the public key is large enough, only someone with knowledge of the prime numbers can practically decode the message. Cryptosystems can be broken by breaking the encryption. The problem is known as symmetric encryption.

System is a relatively slow algorithm, and because of this, it is rarely used to directly encrypt user data. More often, symmetric encryption passes the encrypted shared key to symmetric key cryptography, which in turn can perform multiple encryption-decryption operations at high speed.

ElGamal-Cryptosystem: In cryptography, the **ElGamal**-Cryptosystem is an asymmetric key encryption algorithm for public-key cryptography based on Diffie–Hellman key exchange. It was described by Taher Elgamal in 1985. **ElGamal**-Cryptosystem is used in the free GNU Privacy Guard software, recent versions of PGP, and other cryptosystems. The Digital Signature Algorithm is a variant of the **ElGamal**-Cryptosystem signature scheme, not to be confused with **ElGamal**-Cryptosystem. **ElGamal** encryption can be defined over any cyclic group G, such as the multiplicative group of integers modulo n. Its security depends on the difficulty of a certain problem in G related to computing discrete logarithms. Along with **RSA**, other public-key cryptosystems have been proposed. Many of them are based on different versions of the discrete logarithm problem. **ElGamal** cryptosystem, called

Elliptic Curve Variation is based on the discrete logarithm problem. It gains strength from the assumption that the discrete logarithm cannot be found in a practical time limit for a given number, while the inverse operation of a power can be computed efficiently.

We now consider a simple version of **ElGamal** which deals with modulosic numbers. In the case of elliptic curves, it is based on various number systems.

C program for the **ElGamal** algorithm:

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
int e1, e2;
int p, d;
int C1, C2;
FILE *out1, *out2;
int gcd(int a, int b)
{
int q, r1, r2, r;
if (a > b) {
r1 = a;
r2 = b;
}
```

```
else {
r1 = b;
r2 = a;
}
while (r2 > 0) {
q = r1 / r2;
r = r1 - q * r2;
r1 = r2;
r2 = r;
}
return r1;
}
int PrimarityTest(int a, int i)
{
int n = i - 1;
int k = 0;
int m, T;
while (n % 2 == 0) {
k++;
n = n / 2;
}
m = n;
T = FindT(a, m, i);
```

```
if (T == 1 || T == i - 1)
return 1;
for (int j = 0; j < k; j++) {
T = FindT(T, 2, i);
if (T == 1)
return 0;
if (T == i - 1)
return 1;
}
return 0;
}
int PrimitiveRoot(int p)
{
int flag;
for (int a = 2; a < p; a++)
{
flag = 1;
for (int i = 1; i < p; i++) {
if (FindT(a, i, p) == 1 && i < p - 1) {
flag = 0;
}
else if (flag && FindT(a, i, p) == 1 && i == p - 1) {
return a;
```

```
}
}
}
}
int KeyGeneration()
{
do {
do
p = rand() + 256;
while (p % 2 == 0);
} while (!PrimarityTest(2, p));
p = 107;
e1 = 2;
do {
d = rand() % (p - 2) + 1; // 1 <= d <= p-2
} while (gcd(d, p) != 1);
d = 67;
e2 = FindT(e1, d, p);
}
int FindT(int a, int m, int n)
{
int r;
int y = 1;
```

```
while (m > 0)
{
r = m % 2;
FastExponention(r, n, &y, &a);
m = m / 2;
}
return y;
}
int FastExponention(int bit, int n, int* y, int* a)
{
if (bit == 1)
*y = (*y * (*a)) % n;
*a = (*a) * (*a) % n;
}
int Encryption(int Plaintext)
{
out1 = fopen("cipher1.txt", "a+");
out2 = fopen("cipher2.txt", "a+");
int r;
do {
r = rand() % (p - 1) + 1; // 1 < r < p
}
while (gcd(r, p) != 1);
```

```
C1 = FindT(e1, r, p);
C2 = FindT(e2, r, p) * Plaintext % p;
fprintf(out1, "%d ", C1);
fprintf(out2, "%d ", C2);
fclose(out1);
fclose(out2);
}
int Decryption(int C1, int C2)
{
FILE* out = fopen("result.txt", "a+");
int decipher = C2 * FindT(C1, p - 1 - d, p) % p;
fprintf(out, "%c", decipher);
fclose(out);
}
int main()
{
FILE *out, *inp;
// destroy contents of these files (from previous runs, if any)
out = fopen("result.txt", "w+");
fclose(out);
out = fopen("cipher1.txt", "w+");
fclose(out);
out = fopen("cipher2.txt", "w+");
```

```
fclose(out);
KeyGeneration();
inp = fopen("plain.txt", "r+");
if (inp == NULL) {
printf("Error opening Source File.\n");
exit(1);
}
while (1)
{
char ch = getc(inp);
if (ch == EOF)
break; // M < p
Encryption(toascii(ch));
}
fclose(inp);
fclose(out);
FILE *inp1, *inp2;
inp1 = fopen("cipher1.txt", "r");
inp2 = fopen("cipher2.txt", "r");
int C1, C2;
while (1)
{
int ret = fscanf(inp1, "%d", &C1);
```

```
fscanf(inp2, "%d", &C2);
if (ret == -1)
break;
Decryption(C1, C2);
}
fclose(inp1);
fclose(inp2);
return 0;
}
```

Elliptic-curve cryptography (Elliptic-curve Cryptography) :

Elliptic-curve cryptography (Elliptic-curve Cryptography) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields.

Elliptic-curve cryptography requires a smaller key than non-elliptic-curve cryptography to provide equivalent security. Elliptic curves are applied to digital signatures, pseudo-random generators, and other tasks. Indirectly, they can be used for encryption by combining key agreement with a symmetric encryption scheme. They are also used in many integer factorization algorithms based on elliptic curves that have applications in cryptography, such as Lenstra elliptic-curve factorization.

Public-key cryptography is based on the innateness of certain mathematical problems. Early public-key systems are secure by assuming that it is difficult to construct a large integer composed of two or more prime factors. For elliptic-curve-based protocols, it is

assumed that there is a publicly known base It is possible to find the discrete logarithm of a random elliptic curve element with respect to a point: this is the elliptic curve discrete logarithm problem.

The properties and functions of elliptic curves have been studied in mathematics for over 150 years. Their use within cryptography was first proposed in 1985 (separately) by Neal Koblitz from the University of Washington and Victor Miller at IBM. An elliptic curve is not an ellipse (oval shape), but is represented as a looping line crossing two axes (lines on a graph used to indicate the position of a point). Elliptic curves are based on the properties of a special type of equation constructed from a mathematical group (a group of values for which operations can be performed on any two members of the group to produce a third member) called the curve generated by the points where the line intersects the axes. Multiplying a curve at one point by a number will produce another point on the curve, but it is very difficult to figure out which number was used, even if you know the original point and the result.

Equations based on elliptic curves have a feature that is very valuable for cryptography purposes: they are relatively easy to perform, and extremely difficult to inverse. Elliptic curve cryptography is a term used to describe a suite of cryptographic tools and protocols whose security is based on special versions of the discrete logarithm problem that do not use modulo numbers.

Chapter 4

Hash Function

Cryptography Hash Function:

A cryptographic hash function is a hash function suitable for use in cryptography. It is a mathematical algorithm that maps data of arbitrary size (often messages) to a bit string of a fixed size (hash value, hash or message digest) and is one-way, that is, a function that is practically invertible to the inverter. Ideally, the only way to find a message that produces a given hash is to attempt a brute-force search of possible inputs to see if they produce a match, or use a rainbow table of matched hashes. Cryptographic hash functions are a fundamental tool of modern cryptography. Cryptographic hash functions are implemented in information security to evaluate data integrity, authentication controls, and other security mechanisms. Cryptographic hash functions work by generating a checksum value of a data object. If the data is intentionally or unintentionally modified, the checksum value is changed. Thus, the integrity of a data object can be evaluated by comparing and verifying the previous and current checksums.

Cryptographic hash functions should exhibit pre-image resistance, second pre-image resistance and collision resistance properties to ensure resilience against any cryptoclastic attack.

Hash functions are extremely useful and appear in almost all information security applications. A hash function is a mathematical function that converts a numerical input value into another compressed numerical value. The input to a hash function is of arbitrary length, but the output is always of fixed length. The set of values returned by a hash function is called the message digest or simply the hash value.

Features of Hash Function:

Features of hash function are as follows:

1. Output-hash value of fixed length:

Hash function compresses data of any length to a fixed length. This process is often called data hashing.

In general, the hash input is much smaller than the data, so hash function is sometimes called compression function.

Since hash is a smaller representation of a larger data, it is also known as a digest. Hash functions with n-bit output are referred to as n-bit hash functions. Popular hash functions generate values between 160 and 512 bits.

2. Efficiency of operation:

Generally, for any hash function i with input x, computing i(x) is a fast operation.

Computing hash functions is very fast as compared to symmetric encryption.

Properties of Hash Function:

The ideal cryptographic hash function has the following five main properties:

It is deterministic, which means that the same message always results in the same hash.

It is quick to calculate the hash value for any message.

It is practically infeasible to generate a message that gives a given hash value.

A small change made to a message should change the hash value so much that the new hash value appears uncorrelated with the old hash value.

It is possible to find two different messages with the same hash value. To be an effective cryptographic tool, a hash function must have the following properties:

Computationally efficient:

First and foremost, a hash function must be computationally efficient. That is, a computer must be able to perform the mathematical formulation of the hash function in an extremely short time.

This property is probably somewhat obvious. If an ordinary computer needed several minutes to process a cryptographic hash function and get the output, it would not be very practical. To be useful, a hash function must be computationally efficient.

In fact, this is not as big a concern as it was 40 or 50 years ago. Currently, an average home computer can process an advanced hash function in a fraction of a second.

Deterministic: A cryptographic hash function must be deterministic. In other words, for any given input, the hash function must always produce the same result.

If we enter the same input ten million times in a row, then a hash function must produce the same exact output ten million times.

This may even be rather obvious. If a cryptographic hash function were to produce different outputs each time the same input was entered, then the hash function would be random and therefore useless. It would be impossible to verify a specific input, which

is the whole point of a hash function - to be able to verify that an individual digital signature is authentic without having access to the individual key.

Pre-image resistance: This property means that the hash function must be computationally difficult to invert. In other words, if the hash function produces a hash value y, then it must be a difficult process to obtain any input value c that is mirrored on y. This property protects against an attacker who only has a hash value and is trying to find the input.

Applications of Hash Functions:

Hashing provides constant time search, insert and delete operations on average. This is why hashing is one of the most commonly used data structures,

Its applications include problems like finding instances, isolates, counting frequencies of objects, duplicate detection, etc.

There are many other applications of hashing including in modern-day cryptography. Some of these applications are listed below:

- Message Digest
- Password Verification
- Data Structures (Programming Languages)

- Compiler Operations
- Rabin-Karp Algorithm
- Combining File Name and Path Together

Message Digest:

This is an application of cryptographic hash functions. Cryptographic hash functions are functions that produce an output from which it is close to impossible to reach the input. This property of hash functions is called immutability.

For example, suppose we have to store our files on any of the available cloud services. We have to ensure that the files stored by us are not tampered with by any third party. We do this by calculating the hash of that file using a cryptographic hash algorithm. One of the common cryptographic hash algorithms is SHA 256. The maximum size of the hash thus calculated is 32 bytes. So computing the hash of a large number of files will not be a problem. We save these hashes on our local machine. Now, when we download files, we again calculate the hash. Then we match it with the previous hash calculation. So, we know whether our files were tampered or not. If anyone tampers with the file, the hash value of the file will definitely change. It is nearly impossible to tamper with the file without changing the hash.

Password Verification:

Cryptographic hash functions are commonly used in password verification. Let us understand this using an example.

When you access any online website that requires a user login, we enter our e-mail and password to verify that the account we are trying to use is ours. When the password is entered, a hash of the password is calculated which is then sent to the server for verification of the password. The passwords stored on the server are actually calculated hashes of the original password. This is done to ensure that when the password is sent from the client to the server, there is no attacker to intercept it.

Password verification usually relies on cryptographic hashes. Storing all user passwords as plain-text can lead to a massive security breach if the password file is compromised. One method of mitigating this threat is to store only the hash digest of each password. To authenticate a user, the user-submitted password is hashed and compared to the stored hash. A password reset method is required when password hashing. The original password cannot be recalculated from the stored hash value. Standard cryptographic hash functions are designed to be calculated quickly, and, as a result, it is possible to try guessed passwords at a high rate.

Data Integrity Testing:

Data integrity testing is the most common application of hash functions. It is used to generate checksums on data files. This application assures the user about the correctness of the data.

Integrity testing helps the user to detect any changes made to the original file. However, it does not give any assurance about the originality. The attacker, instead of modifying the file data, can change the entire file and calculate all the new hashes and send it to the recipient. This integrity testing application is useful only when the user is sure about the originality of the file.

Functions with these properties are used as hash functions for various purposes, not only in cryptography. Practical applications include message integrity testing, digital signatures, authentication, and various information security applications.

A hash function takes a string of any length as input and generates a fixed length string that acts as a signature for the provided data.

Thus, a person knowing the hash value is unable to know the original message, but only a person who knows the original message can deduce the hash value that can be created from that message. A cryptographic hash function should behave as much as possible like a random function, while being deterministic and efficiently computable. A

cryptographic hash function is considered insecure from a cryptographic point of view if any of the following are computationally possible:

1. Finding a (previously unseen) message that matches a given hash value.
2. Finding a collision, in which two different messages have the same hash value.

An attacker who can find any of the above computations can use them to replace an authorized message with an unauthorized one.

Ideally, it should be impossible to find two different messages whose digests (hash values) are the same. Additionally, one would not expect an attacker to learn anything useful about a message from its digest (hash value). Of course the attacker learns at least one piece of information by which the attacker can identify whether the same message came again.

Chapter 5

Digital Signature

Cryptography Digital Signature:

Digital signature or signature is the oldest public-key form of message authentication. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind the message to the signer.

Similarly, digital signature is a technique that binds a person/entity to digital data. This binding can be independently verified by the recipient as well as any third party.

A digital signature is a cryptographic value calculated from data and a secret key known only to the signer. In the real world, the recipient of a message needs assurance that the message belongs to the sender and should not be able to disprove the origin of the message. This requirement is very important in business applications, as the possibility of dispute over the exchanged data is very high. A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document. The digital equivalent of a handwritten signature or a stamped seal, a digital signature offers far greater inherent security, and is intended to solve the problem of tampering and impersonation in digital communications. Digital signatures can

provide additional assurances of evidence of the origin, identity and status of an electronic document, transaction or message and acknowledge informed consent by the signer.

In many countries, including the United States, digital signatures are considered to be as legally binding as traditional document signatures. The United States Government Publishing Office publishes electronic versions of the budget, public and private laws, and congressional bills with digital signatures.

How do digital signatures work?

Digital signatures are based on public key cryptography, also known as asymmetric cryptography. Using a public key algorithm, such as MTTR, can generate two keys that are mathematically linked: a personal and a public one. Digital signatures work because public key cryptography relies on two mutually authenticating cryptographic keys. The person creating the digital signature uses his or her personal key to encrypt the signature-related data, the only way to decrypt that data is with the signer's public key. This is how digital signatures are authenticated. Digital signature technology requires all parties to trust that the person creating the signature is able to keep his or her personal key secret. If someone else has access to the signer's personal key, that party can create a fraudulent digital signature in the name of the private key holder. How to Create a Digital Signature? To create a digital signature, signing software, such as an email program, creates a one-way hash of the electronic data to be signed.

The individual key is then used to encrypt the hash.

The encrypted hash, along with other information, such as the hashing algorithm,

is the digital signature. The reason for encrypting the hash instead of the entire message or document is that a hash function can convert a desired input into a fixed length value, which is usually very small. This saves time because hashing is much faster than signing.

The value of the hash is unique to the hashed data. Any change to the data, even a change to a single character, will result in a different value. This feature enables other people to validate the integrity of the data by using the signer's public key to decrypt the hash. If the decrypted hash matches another calculated hash of the same data, it proves that the data has not changed since it was signed. If the two hashes do not match, the data has been tampered with in some way. The integrity or signature was created with a private key that does not correspond to the public key presented by the signer authentication. A digital signature can be used with any type of message, whether it is encrypted or not. Only so the recipient can be sure of the sender's identity and that the message remained intact. Digital signatures make it difficult for a signer to deny signing something, assuming their personal key has not been compromised. Since a digital signature is unique to both the document and the signer and binds them together, this property is called non-repudiation. Digital signatures should not be confused with digital certificates. A digital certificate, an electronic document containing the digital signature of the issuing certificate authority, binds a public key to an identity and can be used to verify that a public key belongs to a particular person or entity. Most modern email programs support the use of digital signatures and digital certificates,

making it easy to sign any outgoing email and digitally sign incoming messages. Digital signatures are used extensively to provide proof of authenticity, data integrity and non-repudiation of communications and transactions conducted over the Internet.

Digital Signature vs. Electronic Signature:

Digital signature is a technical term, defining the result of a cryptographic process that can be used to authenticate a sequence of data, electronic signature or e-signature is a legal term that has been legislatively defined. For example, in the United States, the term was defined as electronic signatures in the Global and National Commerce Act, passed in 2000, to mean

an electronic sound, symbol, or process that is associated with or logically connected with a contract or other record and executed or adopted by a person with the intent of signing the record.

This means that any digital signature that can be expressed digitally in electronic form and associated with a representation of a record can be a type of electronic signature.

More commonly, however, an electronic signature can be as simple as the signer's name being entered into a form on a webpage.

To be considered valid, electronic signature schemes must include the following three things:

- A method of verifying the identity of the entity performing the signing.
- It is a method of verifying that the signing entity has signed the document.
- It is a method of verifying that the electronic signature is actually associated with the signed document.

Authenticated digital (digital) signatures provide cryptographic proof that a document was signed by the signing entity and that the document has not been altered. Not all electronic signatures can provide the same guarantee.

Any process of authentication protects two parties against a third party.
However, this process does not protect the parties against each other. This means that in situations where there is not complete trust between the sender and the recipient, something more than authentication is required. This problem can be solved by using digital (digital) signatures. A digital (digital) signature is analogous to a handwritten signature and confirms the author, date and time of the signature. The signature should also be capable of authenticating the content at the time of signing.

The main requirements of a digital signature are:

- It is unique to the sender.
- It should be recognizable and verifiable.

There are various approaches to digital signatures, which broadly fall into two categories – direct and intermediary.

Direct Digital Signature:

A direct digital signature can be created by encrypting the entire message with the sender's personal key or by encrypting the hash value of the message with the sender's personal key. The following figure explains the process of creating a digital signature. The output is called the digital signature and is attached to the message. To verify the signature, the recipient inputs the message, the signature and the sender's public key. If the result is consistent, the signature is considered authentic. Otherwise, the signature is considered forged or the message has been tampered with. This is because the calculated value is based on the signature and the content of the message. Any change in the values of the digital signature or the message content is the result of a mismatch between the calculated value and the received value. This indicates that either the signature has been forged or the message content has been modified.

Direct digital signature scheme has one drawback - the whole scheme depends on the validity of the sender's personal key. If the sender denies responsibility that he has sent the message and claims that the personal key has been lost or compromised, then someone has to forge the signature.

Arbitrary Digital Signature:

Arbitrary digital signature scheme is used to overcome the non-repudiation problem faced in direct digital signature. In this scheme, every signed message of the sender, which is sent to the recipient, first goes to an intermediary who tests about the origin and content of the signature. The message is dated and sent to the recipient. The presence of the intermediary solves the problem of the sender forging the signature. For example, when X sends a digitally signed message to X, an intermediary first verifies X's signature. After the signature is validated, the message is then sent to X with the date of verification and a notice that the signature belongs to X. The following are the steps in creating a digital signature:

1. The message digest is completed by applying a hash function to the message and then the message digest is encrypted using the sender's personal key to create a digital signature. (Digital signature = encryption (personal key of sender, message digest) and message digest = message digest algorithm (message).

2. The digital signature is then transmitted along with the message. (Message $ digital signature is transmitted)

3. The recipient decrypts the digital signature using the sender's public key (This assures authenticity, as only the sender has his personal key so only the sender can encrypt using his personal key which can thus be decrypted by the sender's public key).

4. The recipient now has the message digest.

5. The recipient can calculate the message digest from the message (the actual message is sent along with the digital signature).

6. The receiver calculates the message digest and the message digest (message digest) The message digest (obtained by decryption on the digital signature) must be the same to ensure integrity.

The message digest is calculated using a one-way hash function, i.e. a hash function in which the hash value is easy to calculate, but the exact hash value is extremely difficult to calculate.

Format of Digital Signature:

As mentioned earlier, the digital signature scheme is based on public key cryptography. The following points explain the whole process in detail -
Every individual adopting this scheme has a public-personal key pair.

Usually, the key pairs used for encryption/decryption and signing/verification are separate. The personal key used for signing is referred to as the signing key and the public key as the verification key.

The signer feeds the data to the hash function and generates the hash of the data. The hash value and the signature key are then fed to the signature algorithm which produces a digital signature at the given hash. The signature is attached to the data and then both are sent to the verifier.

The verifier enters the digital signature and verification key into the verification algorithm. The verification algorithm returns some value as output.

The verifier also runs the same hash function on the received data to generate a hash value.

For verification, this hash value and the output of the verification algorithm are compared. Based on the comparison result, the verifier decides whether the digital signature is valid or not.

Importance of Digital Signature:

Digital signatures using public key cryptography are considered to be very important and useful tools for achieving information security.

In addition to the ability to provide non-repudiation of a message, digital signatures also provide message authentication and data integrity. Now we will briefly see how this is achieved by digital signatures -

Message Authentication: When a verifier validates a digital signature using a sender's public key, he is assured that the signature has been created only by the sender who has the corresponding secret private key and no one else.

Data Integrity: If an attacker has access to the data and modifies it, digital signature verification fails at the recipient's end. The hash of the modified data will not match the output provided by the hash and verification algorithm. Hence, the receiver can safely reject the message assuming that data integrity has been violated.

Non-repudiation: Since it is assumed that only the signer has knowledge of the signature key, only he can create unique signatures on a given data. Thus the receiver can present the data and digital signature as evidence to third parties if any dispute arises in future.

By adding public-key encryption to the digital signature scheme, we can create a cryptosystem that can provide the four essential elements of security – confidentiality, authentication, integrity and non-repudiation.

Encryption with Digital Signature - In many digital communications, it is desirable to exchange simple encrypted messages to achieve confidentiality. In public key encryption schemes, a public key of the sender is available in the open domain, and hence anyone can spoof the identity of the sender and send any encrypted message to the receiver. It is necessary for users to obtain digital signatures for encryption as well as encrypted data to assure message authentication and non-repudiation. This can be achieved by combining digital signatures with encryption schemes. We will now briefly discuss how to achieve this requirement. There are two possibilities, signature followed by encryption or encryption followed by signature.

However, the cryptosystem based on signature followed by encryption can be used by the receiver to spoof the sender's identity and send that data to a third party. Hence, this method is not preferred. The encryption followed by signature is more reliable and widely adopted. The receiver, after receiving the encrypted data and signature on it, first verifies the signature using the sender's public key. After ensuring the validity of the signature, he retrieves the data through decryption using his personal key. Encrypting a document:

To use digital signature or encryption we need to have a digital identity which is also known as a digital certificate. A digital identity/digital certificate used to do two things. First, it can be used to encrypt emails or files so that they can only be read by the person for whom it is encrypted. Second, it can be used to sign or put a digital signature

on a document to assure that it comes in the same state as it was originally and that no one has added or changed things.

There is a public and a personal key. Our public key is shared with everyone. Our personal key is kept personal. These keys are text documents that appear to be random numbers and letters, but with the proper algorithm, these numbers and letters have a very unique property.

If we run a document through an algorithm with the public key, we get back an encrypted document or an encrypted email.

The public key cannot be used to decrypt the document. This process is a method so it doesn't matter if other people have the public key, they can't read the document.

We must have the personal key to decrypt the document. If we give the encrypted document to the algorithm with the personal key, we will get the original document back.

Public Key Infrastructure:

The most distinctive feature of Public Key Infrastructure (PKI) is that it uses a pair of keys to achieve the underlying security service. The key pair consists of a private key and a public key.

Since public keys are in the open domain, they are prone to misuse.

Thus, it is necessary to establish and maintain some kind of trusted infrastructure to manage these keys. Key Management:

The security of any cryptosystem depends on how securely its keys are managed. Without secure procedures for handling cryptographic keys, the benefits of using strong cryptographic schemes are potentially lost.

It has been observed that cryptographic schemes are rarely compromised through flaws in their design. However, they are often compromised through poor key management.

There are some important aspects of key management which are as follows:

Cryptographic keys are nothing but special pieces of data. Key management refers to the secure administration of cryptographic keys.

Public key assurance: In public key cryptography, public keys are in the open domain and are viewed as public parts of data. By default there is no assurance whether the public key is correct, with whom it can be associated, or what it can be used for. Thus key management of public keys needs to be more clearly focused on the assurance of the purpose of the public key.

The most important requirement of ‘assurance’ of public keys can be achieved through PKI, which is a key management system to support public-key cryptography. PKI assures the public keys. It provides the identity of the public keys and their distribution. The structure of PKI includes the following components:

- Public key certificates, commonly called digital certificates.
- Personal key tokens
- Certification authorities.

- Registration authorities.
- Certificate management system.

PKI is a set of roles, policies and procedures required to create, manage, distribute, use, store and modify digital certificates and to manage public-key encryption. PKI is intended to facilitate secure electronic transfer of information for a wide range of network activities such as e-commerce, Internet banking and confidential email. This is necessary for activities where simple passwords are an inadequate authentication method and more rigorous proof is needed to confirm the identity of the parties involved in the communication and to validate the information being transferred.

In cryptography, a binding is a mechanism that binds a public key to the identity of an entity (such as people and organizations). The binding is established through a process of registration and issuance of certificates by a certificate authority (CA). Depending on the assurance level of the binding, this can be done as an automated process or under human supervision. The role of the CA that assures valid and correct registration is called the registration authority (RA).

The CA is responsible for accepting requests for digital certificates and authenticating the requesting entity. In Microsoft Q&A, a registration authority is typically called a subordinate authority.

Signing Computer Programs:

Digital signatures can also be used to authenticate software applications. The manufacturer of a computer program can generate a digital signature for the executable. When a user downloads the program, he can verify that the digital signature is correct. Then he knows that this program was indeed created by that particular manufacturer. If he trusts that manufacturer, he can safely install the application. Of course the manufacturer assures that the application will not do anything malicious.

Digital signature certificates are helpful in authenticating the details of the individual holder's personal information when conducting business online.

Reduced Cost and Time: Instead of physically signing hard copy documents and scanning them for sending via e-mail, you can digitally sign PDF files and send them very quickly. The holder of a digital signature certificate does not have to be physically present to transact or authorize any business.

Data integrity: Documents that are digitally signed cannot be altered or edited after signing, which makes the data safe and secure.

Government agencies often ask for these certificates to cross-check and verify business transactions.

Authenticity of documents: Digitally signed documents give the recipient the confidence of being assured of the authenticity of the signer. They can take action based on such documents without worrying about forged documents.

The use of digital signatures is similar to that of a handwritten signature. Digital signatures use digital keys to authenticate a person. A digital signature certificate attached to any document is a binding commitment by the signature holder as per the IT law in India. The use of digital signatures is getting difficult because unlike handwritten signatures, digital signatures are considered impossible to forge or counterfeit. With technological advancements we are moving away from the world of pen and paper to the electronic age. Now a days most of the government and private tenders are collected electronically only. In India, filing of taxes like income tax, sales tax etc. electronically is mandatory for most. The use of digital signatures in all these electronic transactions is mandatory to make them legally binding. The use of digital signature certificates includes sending secure emails as well as web based monetary transactions. We can use digital signatures for code signing of any software developed by us or to prove ownership of our domain name. The use of digital signatures is also mandatory for patent and trademark registration in India. The use of digital signatures is however not restricted to this. The use of digital signatures also includes banking transactions and mobile security.

Network Security - Critical Need:

Information and efficient communication are the two most important strategic issues for the success of every business. With the advent of electronic means of communication and storage, more and more businesses have shifted to using data networks to communicate, store information and access resources. There are various types and levels of network infrastructure that are used to run businesses.

It can be said that in the modern world nothing has had a greater impact on businesses than the networked computer. But networking brings with it security threats which, if mitigated, allow the benefits of networking to outweigh the risks. Role of Networking in Business:

Nowadays, computer networking is viewed as a resource by almost all businesses. This resource enables them to collect, analyze, organize, and disseminate information that is essential to their profit. Most businesses have established networks to remain competitive.

The most obvious role of computer networking is that organizations can store almost any type of information at a central location and retrieve it at the desired location through the network.

Advantages of Networks:

Computer networking enables people to share information and ideas easily, so they can work more efficiently and productively. Networks improve activities such as purchasing, selling, and customer service. Networking makes traditional business processes more efficient, more manageable, and less costly.

The major benefits that attract businesses from computer networks are as follows:

Resource Sharing - A business can reduce the amount spent on hardware by sharing components and peripheral devices connected to the network.

Streamlined Business Processes - Computer networks enable businesses to streamline their internal business processes.

Collaboration Between Departments - When two or more departments of a business connect selected parts of their network, they can streamline business processes, which normally take an increased amount of time and effort and often face difficulties in achieving higher productivity.

Improved Customer Relations - Networks provide several benefits to customers such as ease of doing business, prompt service response, etc. There are many other business

specific benefits that accrue from networking. Such benefits have made it essential for all types of businesses to adopt computer networking.

Need for Network Security:

With the advancement in modern technology with the increasing capacity of computer networks, threats to wired or wireless networks have increased significantly. The excessive use of the Internet in today's world for various business transactions has led to challenges of information theft and other attacks on business intellectual property.

In the present era, most businesses operate through network applications, and therefore, all networks are vulnerable to attack. The most common security threats to business networks are data interception and theft, and identity theft. Network security is a specialized field that deals with thwarting such threats and providing usability, reliability, integrity, and security of a business's computer networking infrastructure.

Importance of Network Security for Business:

Protection of business assets - This is the primary goal of network security. Assets means information stored in computer networks. Information is as important and valuable as any other tangible asset of a company. Network security deals with the integrity, security, and secure access to confidential information.

Compliance with regulatory requirements - Network security measures help businesses comply with government and industry specific regulations regarding information security.

Secure collaborative work - Network security encourages co-worker collaboration and facilitates communication with customers and suppliers by providing them secure network access.

It increases customer and consumer confidence that their sensitive information is secure. Reduced risk - Adoption of network security reduces the impact of a security breach, including legal action that can bankrupt small businesses. Gaining competitive advantage - Developing an effective security system for the network gives an organization a competitive edge. In the field of Internet financial services and e-commerce, network security is of prime importance. The network needs protection against attackers and hackers. Network security involves two basic securities. The first is the security of data information i.e. to protect the information from unauthorized access and loss and the second is computer security i.e. to protect data and thwart hackers. Here network security does not mean security only in a network or any single network of a network.

Now our network security requirement is divided into two parts. One is the requirement of information security and the other is the requirement of computer security.

On the internet or any network of an organization, thousands of important information is exchanged every day. This information can be misused by attackers.

Information security is required due to the following reasons:

1. To protect the confidential information of users on the internet itself. No other person should see or access it.

2. To protect the information from accidental or intentional unwanted editing by unauthorized users.

3. To protect the information from loss and to ensure its correct delivery to its destination.

4. To manage the acknowledgement of the received message by any node to prevent denial by the sender in specific circumstances. 5. To restrict a user to send some message to another user in the name of a third party. For example, user X creates a message for his own interest containing some friendly instructions and sends it to user Y in such a way that Y accepts the message as coming from the manager of the organization.

6. To avoid unwanted delays in the transmission lines/path in case of urgency, to make it reach the required destination on time.

7. To avoid data packets or information packets getting lost in the network for an infinitely long time and thus fail to catch it due to some internal faults caused by the increasing congestion in the line.

Another part of network security involves computer security. Computer security means to protect our computer system from unwanted damage caused by the network. One of the major reasons of such damage is viruses and spyware which can erase all the information from our hard disk or sometimes they can be destructive enough to cause hardware problems.

Certainly the network must be protected from such harmful software. People who intentionally put such software on the network are called hackers. As network is a part of computer, so computer protection from hackers is also a part of network security. The requirements of computer protection from hackers are as follows:

- It must be protected from replication and capture of viruses from infected files.
- It needs proper protection from worms and bombs.
- Protection from Trojan horses is required as they are quite dangerous for our computer.

The need for email security:

Due to the popularity of email as an attack route, it is critical that enterprises and individuals take measures to secure their email accounts against common attacks as well as attempts to gain unauthorized access to accounts or communications.

Malware sent via email messages can be quite destructive. Phishing emails sent to employees often contain malware in attachments designed to look like legitimate documents or include hyperlinks that lead to malware-creating websites. Opening an email attachment or clicking on a link in an email can be all it takes for accounts or devices to become compromised. Phishing emails can be used to trick recipients into sharing sensitive information, often by impersonating a trusted business or trusted contacts.

Phishing attacks against businesses often target departments that handle sensitive personal or financial information, such as accounts or human resources. In addition to impersonating known vendors or company officials, attackers try to create a sense of urgency in phishing emails to increase their chances of success.

Typically, phishing emails aimed at stealing information will ask recipients to confirm their login information, passwords, social security numbers, bank account numbers, and even credit card information. Some also link to fake websites that look exactly like the

original one in order to trick victims or business partners into entering account or financial information.

E-mail Security Services:

The increasing use of e-mail communication for important transactions demands the provision of some basic security services as follows:

Confidentiality - The e-mail message should not be read by anyone except the intended recipient.

Authentication - The e-mail recipient can be sure of the sender's identity.

Fidelity - The recipient has assurance that the e-mail message has not been altered after it was transmitted by the sender.

Non-repudiation - The e-mail recipient is able to prove to third parties that the sender actually sent the message.

Proof of Submission - The e-mail sender receives confirmation that the message has been handed over to the postal delivery system. Proof of Receipt - The sender gets confirmation that the recipient has received the message.

Security services such as confidentiality, authentication, message integrity and non-repudiation are provided using public key cryptography.

Generally, there are three different scenarios of email communication. We will discuss the ways to achieve the above security services in these scenarios.

Email Security Services Provider:

Email security services are designed to protect company email accounts from unwanted access and mishandling, and secure employee emails from deletion, viruses and theft. Email security providers use threat intelligence to track global cases of ransomware, phishing, spoofing and other email-related cybercrime. Providers use this information to design and maintain firewalls and other equipment, while educating employees on what to avoid and other security measures. Through these combined efforts, email security firms help reduce exposure to corrupted inbound messages and protect sensitive outbound messages.

Chapter 6

Computer Security Concepts

COMPUTER SECURITY The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/ data, and telecommunications).

This definition introduces three key objectives that are at the heart of computer security:

• Confidentiality: This term covers two related concepts: Data1 confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorized individuals.

Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

• Integrity: This term covers two related concepts: Data integrity: Assures that information and programs are changed only in a specified and authorized manner. System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

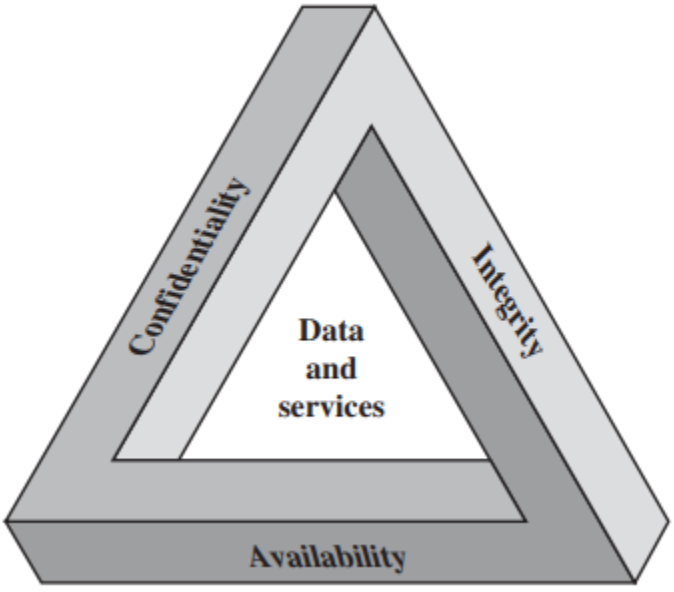
• Availability: Assures that systems work promptly and service is not denied to authorized users.

◆ The Open Systems Interconnection (OSI) security architecture provides a systematic framework for defining security attacks, mechanisms, and services.

◆ Security attacks are classified as either passive attacks, which include unauthorized reading of a message of file and traffic analysis or active attacks, such as modification of messages or files, and denial of service.

◆ A security mechanism is any process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack. Examples of mechanisms are encryption algorithms, digital signatures, and authentication protocols.

◆ Security services include authentication, access control, data confidentiality, data integrity, nonrepudiation, and availability.



The Security Requirements Triad

• Confidentiality: Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information.

• Integrity: Guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information.

• Availability: Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system.

Although the use of the CIA triad to define security objectives is well established, some in the security field feel that additional concepts are needed to present a complete picture.Two of the most commonly mentioned are as follows:

• Authenticity: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.

• Accountability: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action. Because truly secure systems are not yet an achievable goal, we must be able to trace a security breach to a responsible party. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

Examples We now provide some examples of applications that illustrate the requirements just enumerated.

For these examples, we use three levels of impact on organizations or individuals should there be a breach of security (i.e., a loss of confidentiality, integrity, or availability). These levels are defined in FIPS PUB 199:

• Low: The loss could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.A limited adverse effect means that, for example, the loss of confidentiality, integrity, or availability might

(i) cause a degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced;

(ii) result in minor damage to organizational assets;

(iii) result in minor financial loss; or

(iv) result in minor harm to individuals.

• Moderate: The loss could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals. A serious adverse effect means that, for example, the loss might

(i) cause a significant degradation in mission capability to an extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is significantly reduced;

(ii) result in significant damage to organizational assets;

(iii) result in significant financial loss; or (iv) result in significant harm to individuals that does not involve loss of life or serious, life-threatening injuries.

• High: The loss could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals. A severe or catastrophic adverse effect means that, for example, the loss might

(i) cause a severe degradation in or loss of mission capability to an extent and duration that the organization is not able to perform one or more of its primary functions;

(ii) result in major damage to organizational assets;

(iii) result in major financial loss; or

(iv) result in severe or catastrophic harm to individuals involving loss of life or serious, life-threatening injuries.

The Challenges of Computer Security Computer

The Challenges of Computer Security Computer and network security is both fascinating and complex. Some of the reasons follow:

1. Security is not as simple as it might first appear to the novice. The requirements seem to be straightforward; indeed, most of the major requirements for security services can be given self-explanatory, one-word labels: confidentiality, authentication, nonrepudiation, or integrity. But the mechanisms used to meet those requirements can be quite complex, and understanding them may involve rather subtle reasoning.

2. In developing a particular security mechanism or algorithm, one must always consider potential attacks on those security features. In many cases, successful attacks are designed by looking at the problem in a completely different way, therefore exploiting an unexpected weakness in the mechanism.

3. Because of point 2, the procedures used to provide particular services are often counterintuitive.Typically, a security mechanism is complex, and it is not obvious from the statement of a particular requirement that such elaborate measures are needed. It is

only when the various aspects of the threat are considered that elaborate security mechanisms make sense.

4. Having designed various security mechanisms, it is necessary to decide where to use them. This is true both in terms of physical placement (e.g., at what points in a network are certain security mechanisms needed) and in a logical sense [e.g., at what layer or layers of an architecture such as TCP/IP (Transmission Control Protocol/Internet Protocol) should mechanisms be placed].

5. Security mechanisms typically involve more than a particular algorithm or protocol. They also require that participants be in possession of some secret information (e.g., an encryption key), which raises questions about the creation, distribution, and protection of that secret information. There also may be a reliance on communications protocols whose behavior may complicate the task of developing the security mechanism. For example, if the proper functioning of the security mechanism requires setting time limits on the transit time of a message from sender to receiver, then any protocol or network that introduces variable, unpredictable delays may render such time limits meaningless.

6. Computer and network security is essentially a battle of wits between a perpetrator who tries to find holes and the designer or administrator who tries to close them. The great advantage that the attacker has is that he or she need only find a single weakness, while the designer must find and eliminate all weaknesses to achieve perfect security.

7. There is a natural tendency on the part of users and system managers to perceive little benefit from security investment until a security failure occurs.

8. Security requires regular, even constant, monitoring, and this is difficult in today's short-term, overloaded environment.

9. Security is still too often an afterthought to be incorporated into a system after the design is complete rather than being an integral part of the design process.

10. Many users and even security administrators view strong security as an impediment to efficient and user-friendly operation of an information system or use of information.

SECURITY ATTACKS

A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of passive attacks and active attacks. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

Passive Attacks

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis. The release of

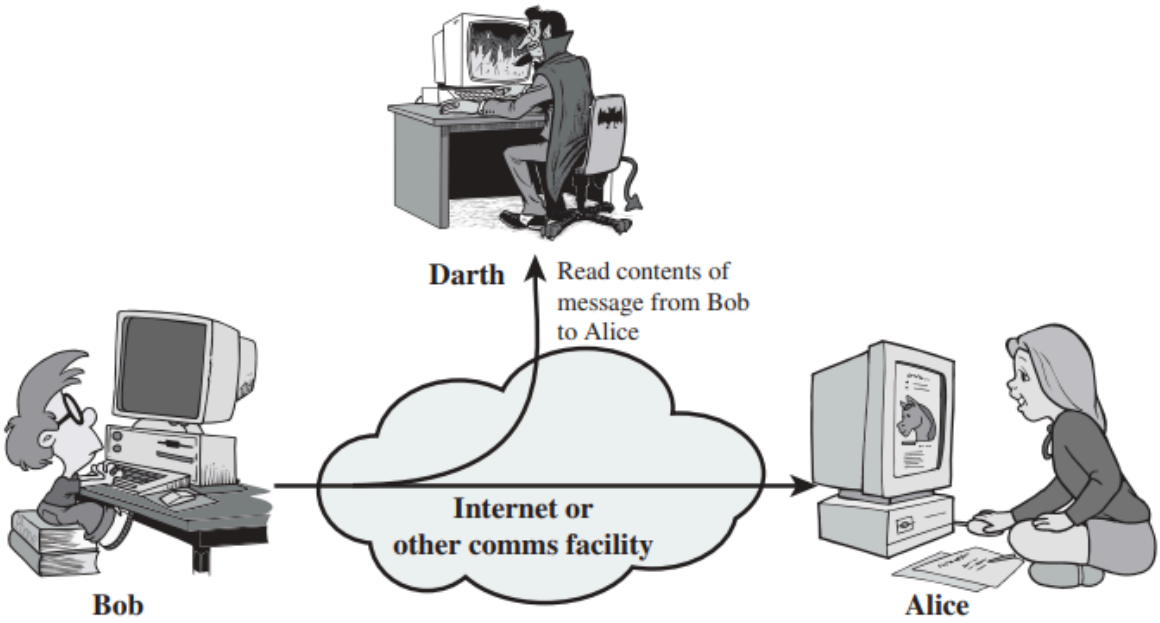
message contents is easily understood (Figure 1.2a).A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information.We would like to prevent an opponent from learning the contents of these transmissions. A second type of passive attack, traffic analysis, is subtler (Figure 1.2b). Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message.

The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location 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 alteration of the data.Typically, the message traffic is sent and received in an apparently normal fashion, and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

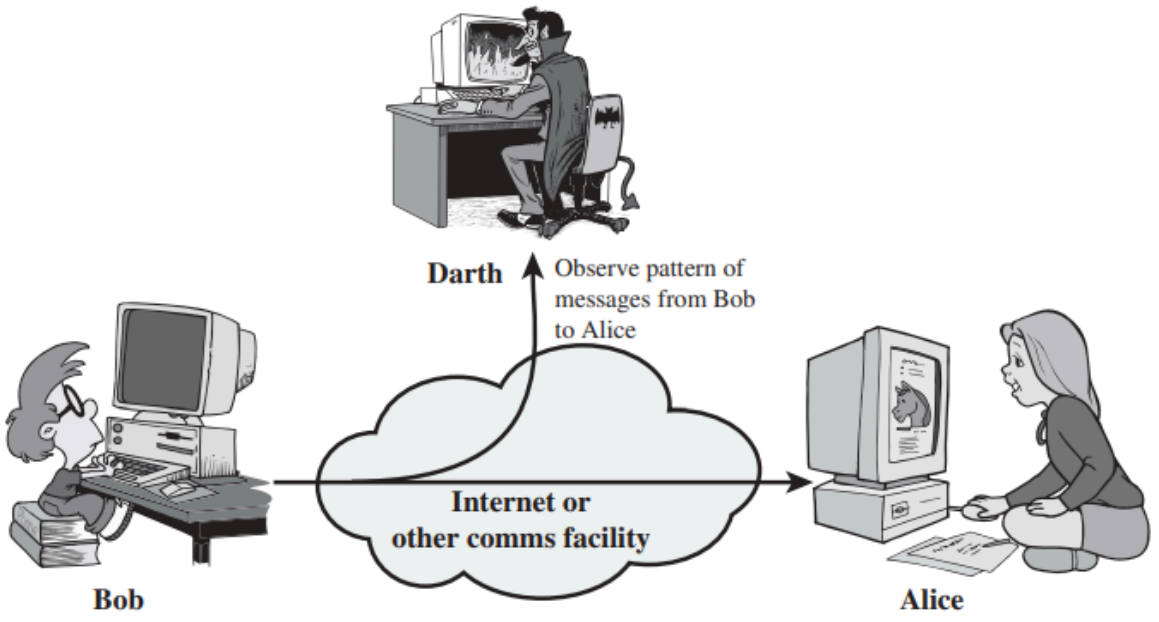
Active Attacks

Active attacks involve some modification of the data 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 entity that has 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. This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination.



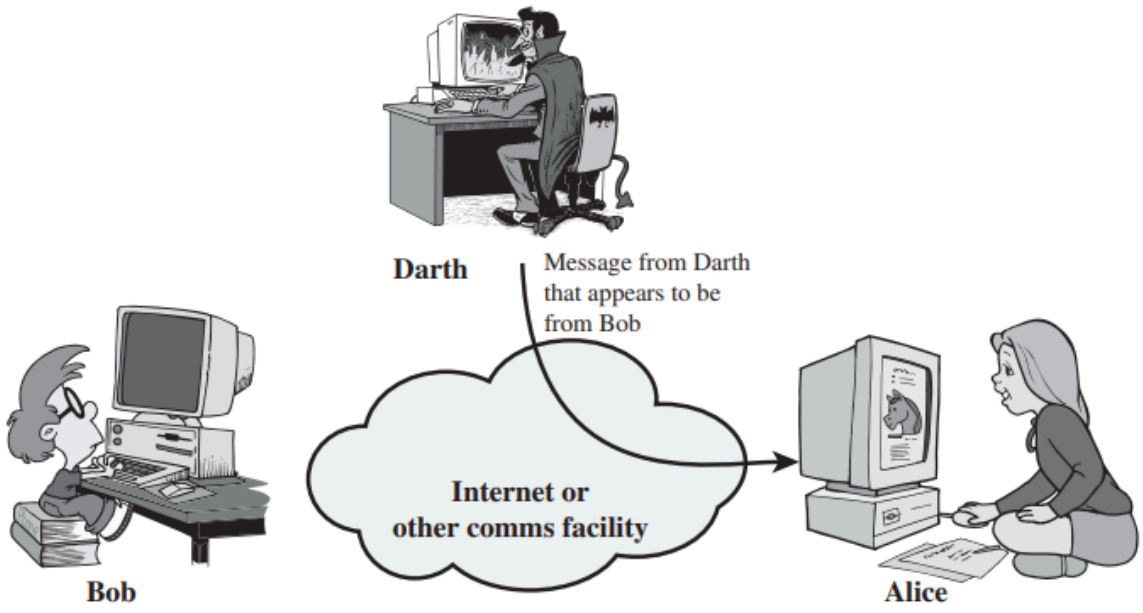
(a) Release of message contents



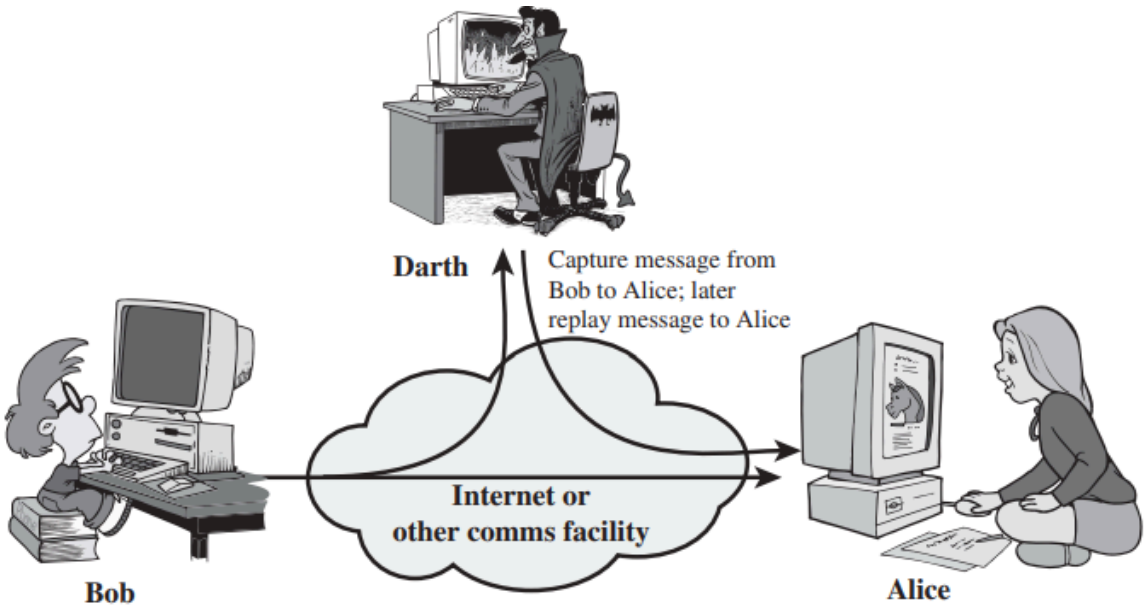
(b) Traffic analysis

(e.g., the security audit service).Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as

to degrade performance. Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success.



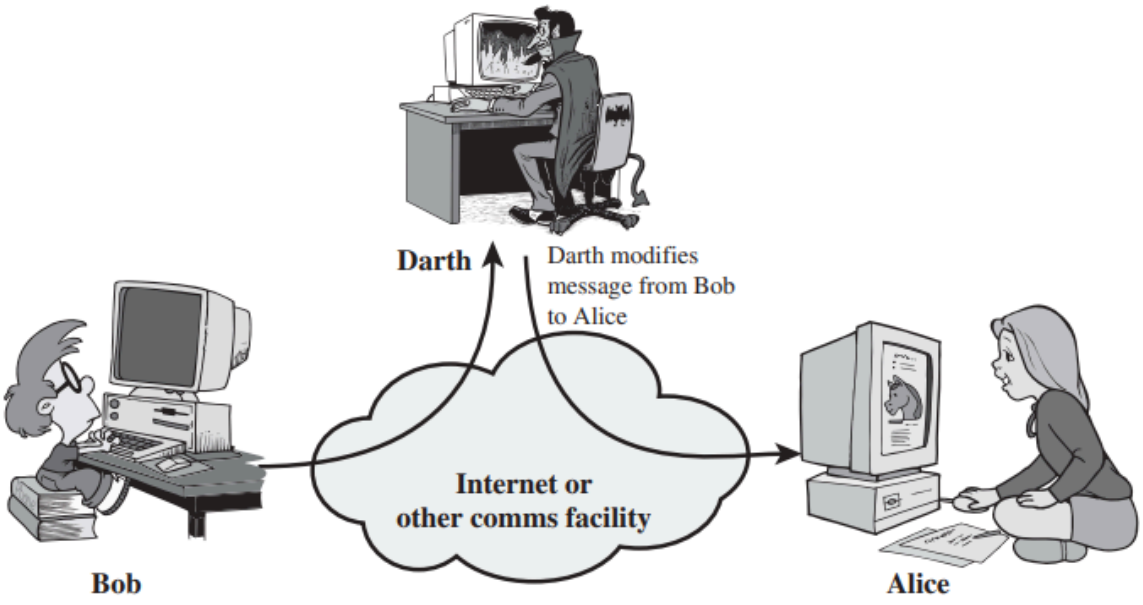
(a) Masquerade



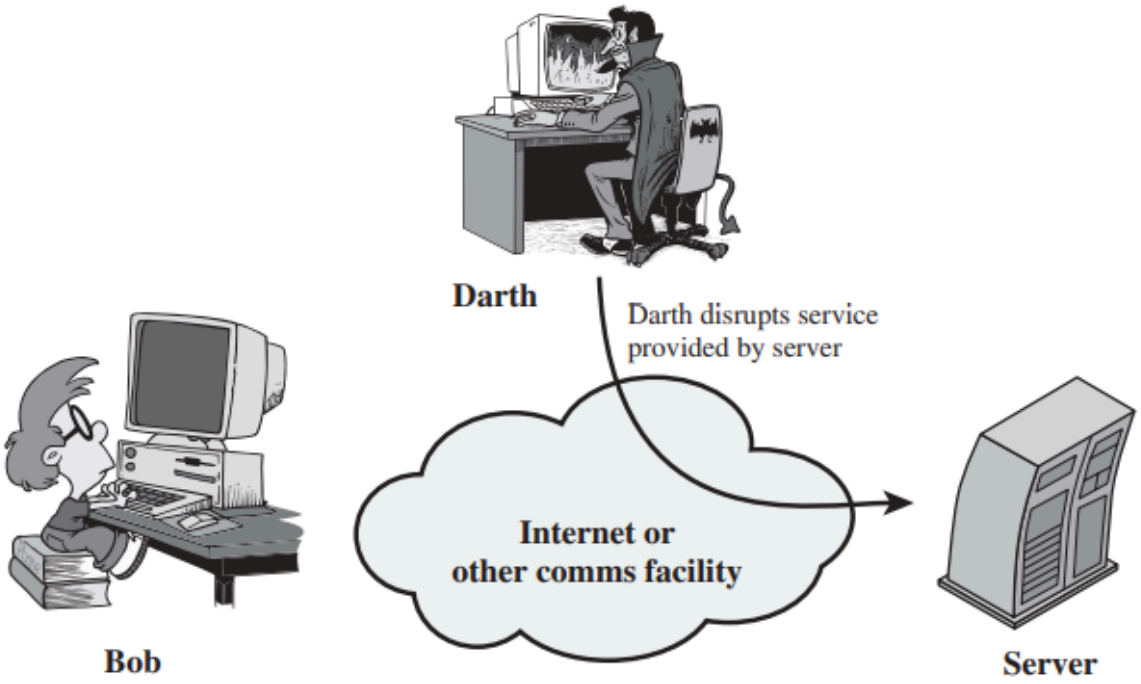
(b) Replay

On the other hand, it is quite difficult to prevent active attacks absolutely because of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal

is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it may also contribute to prevention.



(c) Modification of messages



(d) Denial of service

Authentication The authentication service is concerned 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 of a 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 unauthorized transmission or reception. Two specific authentication services are defined in X.800:

• Peer entity authentication: Provides for the corroboration of the identity of a peer entity in an association. Two entities are considered peers if they implement to same protocol in different systems; e.g., two TCP modules in two communicating systems. Peer entity authentication is provided for use at the establishment of, or at times during the data transfer phase of, a connection. It attempts to provide confidence that an entity is not performing either a masquerade or an unauthorized replay of a previous connection.

• Data origin authentication: Provides for the corroboration of the source of a data unit. It does not provide protection against the duplication or modification of data units. This type of service supports applications like electronic mail, where there are no prior interactions between the communicating entities. Access Control 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, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

Data Confidentiality Confidentiality is the protection of transmitted data from passive attacks. With respect to the content of a data transmission, several levels of protection can be identified.The broadest service protects all user data transmitted between two users over a period of time. For example, when a TCP connection is set up between two systems, this broad protection prevents the release of any user data transmitted over the TCP connection. Narrower forms of this service can also be defined, including

the protection of a single message or even specific fields within a message. These refinements are less useful than the broad approach and may even be more complex and expensive to implement. The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communications facility.

Data Integrity As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message. Again, the most useful and straightforward 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 without regard to any larger context, generally provides protection against message modification only. We can make a distinction between service with and without recovery. Because the integrity service relates to active attacks, we are concerned with detection rather than 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.

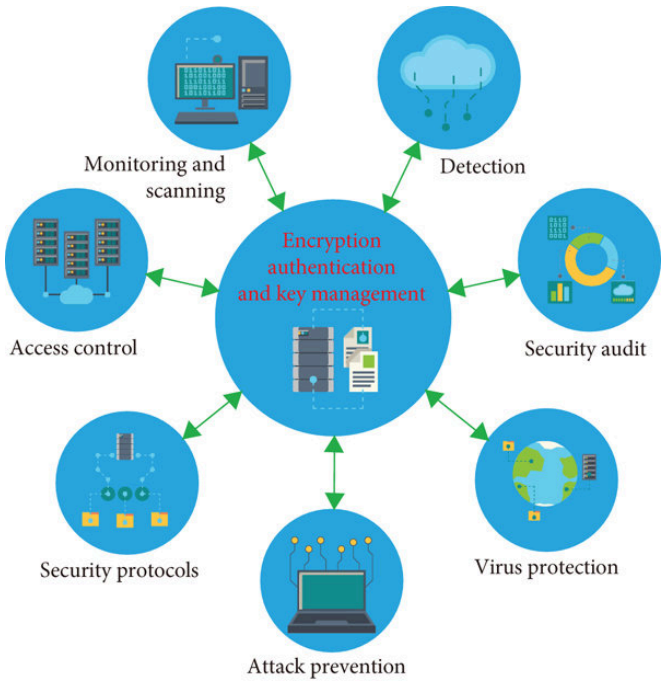
The incorporation of automated recovery mechanisms is, in general, the more attractive alternative.

Nonrepudiation

Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.

Availability Service

Both X.800 and RFC 2828 define availability to be the property of a system or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system (i.e., a system is available if it provides services according to the system design whenever users request them).



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 of a distributed system. X.800 treats availability as a property to be associated with various security services. However, it makes sense to call out specifically an availability service. An availability service is one that protects a system to ensure its availability.This service addresses the security concerns raised by denial-of-service attacks. It depends on proper management and control of system resources and thus depends on access control service and other security services.

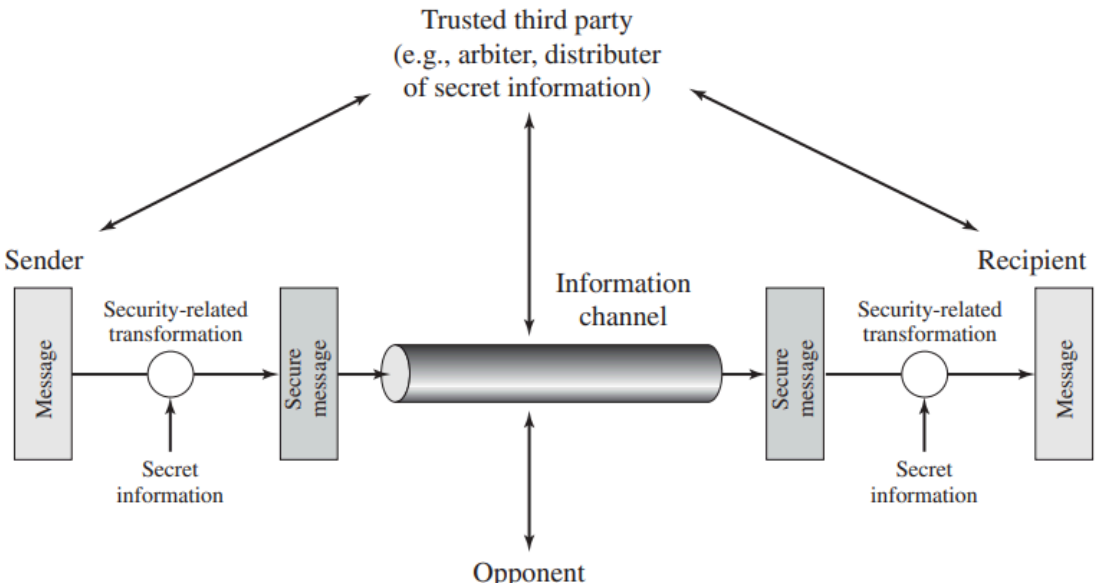
A MODEL FOR NETWORK SECURITY

A model for much of what we will be discussing is captured, in very general terms.

A message is to be transferred from one party to another across some sort of Internet service. The two parties, who are the principals in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) 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 of the message, which scrambles the message so that it is unreadable by

the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.



Model for Network Security

• Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.6 A trusted third party may be needed to achieve secure transmission. For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent. Or a third party may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission. This general model shows that there are four basic tasks in designing a particular security service:

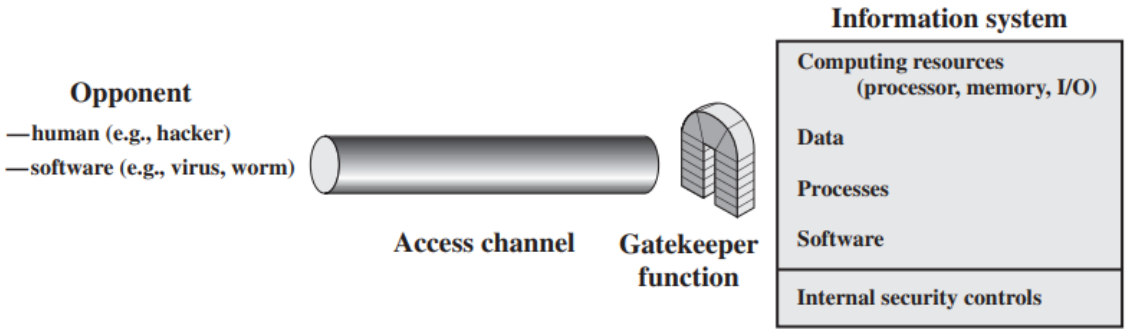
1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.

2. Generate the secret information to be used with the algorithm.

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

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

Parts One through Five of this book concentrate on the types of security mechanisms and services that fit into the model shown in Figure 1.4. However, there are other security-related situations of interest that do not neatly fit this model but are considered in this book.A general model of these other situations is illustrated by Figure 1.5, which reflects a concern for protecting an information system from unwanted access. Most readers are familiar with the concerns caused by the existence of hackers, who 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. 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).



Network Access Security Model

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 compilers. Programs can present two kinds of threats: • Information access threats: Intercept or modify data on behalf of users who should not have access to that data. • Service threats: Exploit service flaws in computers to inhibit use by legitimate users. Viruses and worms are two examples of software attacks. Such attacks can be introduced into a system by means of a disk that contains the unwanted logic concealed in otherwise useful software. They can also be inserted into a system across a network; this latter mechanism is of more concern in network security. The security mechanisms needed to cope with unwanted access fall into two broad categories (see Figure 1.5). 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 detect and reject worms, viruses, and other similar attacks. Once either an unwanted user or unwanted software gains access, the second line of defense consists of a variety of

internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

RECOMMENDED READING AND WEB SITES

[STAL02] provides a broad introduction to both computer and network security. [SCHN00] is valuable reading for any practitioner in the field of computer or network security: It discusses the limitations of technology, and cryptography in particular, in providing security and the need to consider the hardware, the software implementation, the networks, and the people involved in providing and attacking security. It is useful to read some of the classic tutorial papers on computer security; these provide a historical perspective from which to appreciate current work and thinking. The papers to read are [WARE79], [BROW72], [SALT75], [SHAN77], and [SUMM84]. Two more recent, short treatments of computer security are [ANDR04] and [LAMP04]. [NIST95] is an exhaustive (290 pages) treatment of the subject. Another good treatment is [NRC91]. Also useful is [FRAS97].

ANDR04 Andrews, M., and Whittaker, J. “Computer Security.” IEEE Security and Privacy, September/October 2004.

BROW72 Browne, P. “Computer Security—A Survey.” ACM SIGMIS Database, Fall 1972. FRAS97 Fraser, B. Site Security Handbook.

RFC 2196, September 1997. LAMP04 Lampson, B. “Computer Security in the Real World,” Computer, June 2004.

NIST95 National Institute of Standards and Technology. An Introduction to Computer Security: The NIST Handbook. Special Publication 800–12, October 1995.

NRC91 National Research Council. Computers at Risk: Safe Computing in the Information Age. Washington, D.C.: National Academy Press, 1991. SALT75 Saltzer, J., and Schroeder, M.

“The Protection of Information in Computer Systems.” Proceedings of the IEEE, September 1975. SCHN00 Schneier, B. Secrets and Lies: Digital Security in a Networked World. New York: Wiley, 2000. SHAN77 Shanker, K.

“The Total Computer Security Problem:An Overview.” Computer, June 1977. STAL08 Stallings, W., and Brown, L. Computer Security. Upper Saddle River, NJ: Prentice Hall, 2008. SUMM84 Summers, R. “An Overview of Computer Security.”

IBM Systems Journal, Vol. 23, No. 4, 1984. WARE79 Ware, W., ed. Security Controls for Computer Systems. RAND Report 609–1. October 1979. http://www.rand.org/pubs/reports/R609-1/R609.1.html.

Recommended Web Sites:

The following Web sites7 are of general interest related to cryptography and network security:

• IETF Security Area: Material related to Internet security standardization efforts.

• The Cryptography FAQ: Lengthy and worthwhile FAQ covering all aspects of cryptography.

• Tom Dunigan's Security page: An excellent list of pointers to cryptography and network security Web sites.

• Peter Gutmann's home page: Good collection of cryptography material.

• Helgar Lipma's Cryptology Pointers: Another excellent list of pointers to cryptography and network security Web sites.

• Cryptology ePrint archive: Provides rapid access to recent research in cryptology; consists of a collection of unrefereed papers.

• IEEE Technical Committee on Security and Privacy: Copies of their newsletter and information on IEEE-related activities.

• Computer Security Resource Center: Maintained by the National Institute of Standards and Technology (NIST); contains a broad range of information on security threats, technology, and standards.

• Computer and Network Security Reference Index: A good index to vendor and commercial products, FAQs, newsgroup archives, papers, and other Web sites.

• Security Focus: A wide variety of security information, with an emphasis on vendor products and end-user concerns.

• SANS Institute: Similar to Security Focus. Extensive collection of white papers.

Chapter 7

Intruders Intruder Behavior

A significant security problem for networked systems is hostile, or at least unwanted, trespass by users or software. User trespass can take the form of unauthorized logon to a machine or, in the case of an authorized user, acquisition of privileges or performance of actions beyond those that have been authorized. Software trespass can take the form of a virus, worm, or Trojan horse. All these attacks relate to network security because system entry can be achieved by means of a network. However, these attacks are not confined to network-based attacks. A user with access to a local terminal may attempt trespass without using an intermediate network. A virus or Trojan horse may be introduced into a system by means of an optical disc. Only the worm is a uniquely network phenomenon.Thus, system trespass is an area in which the concerns of network security and computer security overlap. Because the focus of this book is network security, we do not attempt a comprehensive analysis of either the attacks or the security countermeasures related to system trespass. Instead, in this Part we present a broad overview of these concerns. This chapter covers the subject of intruders. First, we examine the nature of the attack and then look at strategies intended for prevention and, failing that, detection. Next we examine the related topic of password management.

One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker. In an important early study of intrusion, Anderson [ANDE80] identified three classes of intruders:

• Masquerader: An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account

• Misfeasor: A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges

• Clandestine user: An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider. Intruder attacks range from the benign to the serious.At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there. At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system. [GRAN04] lists the following examples of intrusion:

• Performing a remote root compromise of an e-mail server
• Defacing a Web server • Guessing and cracking passwords
• Copying a database containing credit card numbers

• Viewing sensitive data, including payroll records and medical information, without authorization

• Running a packet sniffer on a workstation to capture usernames and passwords

• Using a permission error on an anonymous FTP server to distribute pirated software and music files

• Dialing into an unsecured modem and gaining internal network access • Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password.

Intruder Behavior Patterns

The techniques and behavior patterns of intruders are constantly shifting, to exploit newly discovered weaknesses and to evade detection and countermeasures. Even so, intruders typically follow one of a number of recognizable behavior patterns, and these patterns typically differ from those of ordinary users. In the following, we look at three broad examples of intruder behavior patterns, to give the reader some feel for the challenge facing the security administrator.

HACKERS

Traditionally, those who hack into computers do so for the thrill of it or for status. The hacking community is a strong meritocracy in which status is determined by level of competence. Thus, attackers often look for targets of opportunity and then share the information with others. A typical example is a break-in at a large financial institution reported in [RADC04]. The intruder took advantage of the fact that the corporate

network was running unprotected services, some of which were not even needed. In this case, the key to the break-in was the pcAnywhere application. The manufacturer, Symantec, advertises this program as a remote control solution that enables secure connection to remote devices. But the attacker had an easy time gaining access to pcAnywhere; the administrator used the same three-letter username and password for the program. In this case, there was no intrusion detection system on the 700-node corporate network. The intruder was only discovered when a vice president walked into her office and saw the cursor moving files around on her Windows workstation.

(a) Hacker

1. Select the target using IP lookup tools such as NSLookup, Dig, and others.
2. Map network for accessible services using tools such as NMAP.
3. Identify potentially vulnerable services (in this case, pcAnywhere).
4. Brute force (guess) pcAnywhere password.
5. Install remote administration tool called DameWare.
6. Wait for administrator to log on and capture his password.
7. Use that password to access remainder of network.

(b) Criminal Enterprise

1. Act quickly and precisely to make their activities harder to detect.
2. Exploit perimeter through vulnerable ports.
3. Use Trojan horses (hidden software) to leave back doors for reentry.

4. Use sniffers to capture passwords.

5. Do not stick around until noticed.

6. Make few or no mistakes.

(c) Internal Threat

1. Create network accounts for themselves and their friends.

2. Access accounts and applications they wouldn't normally use for their daily jobs.

3. E-mail former and prospective employers.

4. Conduct furtive instant-messaging chats.

5. Visit Web sites that cater to disgruntled employees, such as f'dcompany.com.

6. Perform large downloads and file copying.

7. Access the network during off hours.

Intrusion Techniques

The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a back door into the system. Alternatively, the intruder attempts to acquire information that should have been protected. In some cases, this information is in the form of a user password. With knowledge of some other user's password, an intruder can log in to a system and exercise all the privileges accorded to the legitimate user. Typically, a system must maintain a file that associates a password with each authorized user. If such a file is

stored with no protection, then it is an easy matter to gain access to it and learn passwords. The password file can be protected in one of two ways:

• One-way function: The system stores only the value of a function based on the user's password. When the user presents a password, the system transforms that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible) in which the password is used to generate a key for the one-way function and in which a fixed-length output is produced.

• Access control: Access to the password file is limited to one or a very few accounts interviews with a number of password crackers, [ALVA90] reports the following techniques for learning passwords:

1. Try default passwords used with standard accounts that are shipped with the system. Many administrators do not bother to change these defaults.
2. Exhaustively try all short passwords (those of one to three characters).
3. Try words in the system's online dictionary or a list of likely passwords. Examples of the latter are readily available on hacker bulletin boards.
4. Collect information about users, such as their full names, the names of their spouse and children, pictures in their office, and books in their office that are related to hobbies.
5. Try users' phone numbers, Social Security numbers, and room numbers.

6. Try all legitimate license plate numbers for this state. 7. Use a Trojan horse (described in Chapter 21) to bypass restrictions on access. 8. Tap the line between a remote user and the host system.

INTRUSION DETECTION

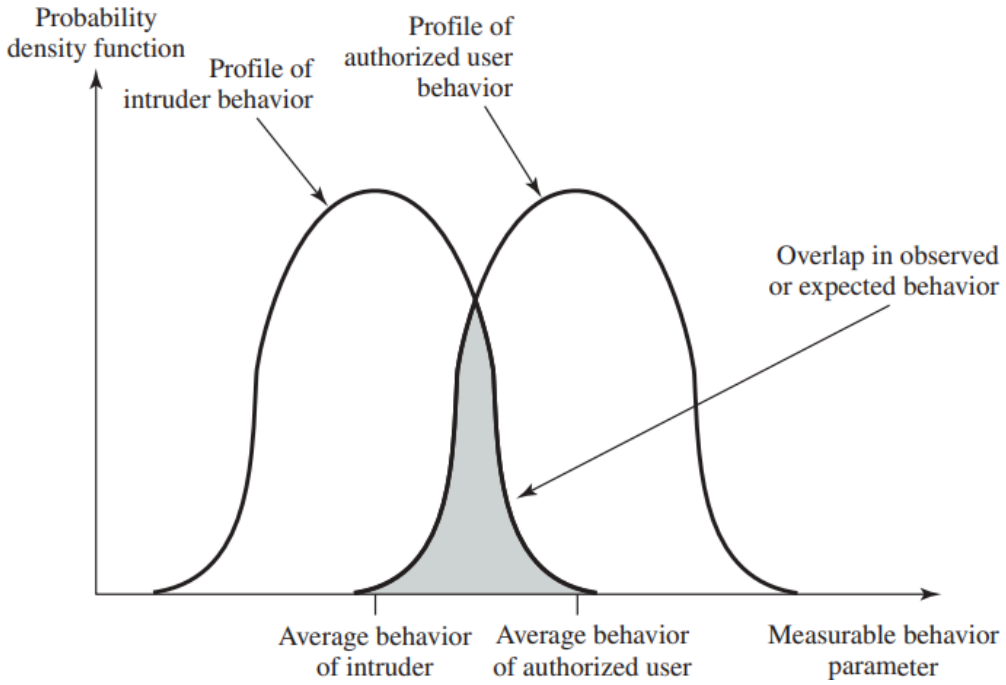
Inevitably, the best intrusion prevention system will fail. A system's second line of defense is intrusion detection, and this has been the focus of much research in recent years. This interest is motivated by a number of considerations, including the following:

1. If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised. Even if the detection is not sufficiently timely to preempt the intruder, the sooner that the intrusion is detected, the less the amount of damage and the more quickly that recovery can be achieved.

2. An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions.

3. Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility. Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified.

Of course, we cannot expect that there will be a crisp, exact distinction between an attack by an intruder and the normal use of resources by an authorized user. Rather, we must expect that there will be some overlap.



Profiles of Behavior of Intruders and Authorized Users

In Anderson's study [ANDE80], it was postulated that one could, with reasonable confidence, distinguish between a masquerader and a legitimate user. Patterns of legitimate user behavior can be established by observing past history, and significant deviation from such patterns can be detected. Anderson suggests that the task of detecting a misfeasor (legitimate user performing in an unauthorized fashion) is more difficult, in that the distinction between abnormal and normal behavior may be small. Anderson concluded that such violations would be undetectable solely through the search for anomalous behavior. However, misfeasor behavior might nevertheless be

detectable by intelligent definition of the class of conditions that suggest unauthorized use.

Finally, the detection of the clandestine user was felt to be beyond the scope of purely automated techniques. These observations, which were made in 1980, remain true today. [PORR92] identifies the following approaches to intrusion detection:

1. Statistical anomaly detection: Involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

a. Threshold detection: This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.

b. Profile based: A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

2. Rule-based detection: Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.

a. Anomaly detection: Rules are developed to detect deviation from previous usage patterns.

b. Penetration identification: An expert system approach that searches for suspicious behavior. In a nutshell, statistical approaches attempt to define normal, or expected, behavior, whereas rule-based approaches attempt to define proper behavior. In terms of the types of attackers listed earlier, statistical anomaly detection is effective against masqueraders, who are unlikely to mimic the behavior patterns of the accounts they appropriate.

On the other hand, such techniques may be unable to deal with misfeasors. For such attacks, rule-based approaches may be able to recognize events and sequences that, in context, reveal penetration. In practice, a system may exhibit a combination of both approaches to be effective against a broad range of attacks.

Audit Records

A fundamental tool for intrusion detection is the audit record. Some record of ongoing activity by users must be maintained as input to an intrusion detection system. Basically, two plans are used:

- Native audit records: Virtually all multiuser operating systems include accounting software that collects information on user activity. The advantage of using this information is that no additional collection software is needed. The disadvantage is that the native audit records may not contain the needed information or may not contain it in a convenient form.

• Detection-specific audit records: A collection facility can be implemented that generates audit records containing only that information required by the intrusion detection system. One advantage of such an approach is that it could be made vendor independent and ported to a variety of systems. The disadvantage is the extra overhead involved in having, in effect, two accounting packages running on a machine. A good example of detection-specific audit records is one developed by Dorothy Denning [DENN87]. Each audit record contains the following fields:

• Subject: Initiators of actions. A subject is typically a terminal user but might also be a process acting on behalf of users or groups of users.All activity arises through commands issued by subjects. Subjects may be grouped into different access classes, and these classes may overlap.

• Action: Operation performed by the subject on or with an object; for example, login, read, perform I/O, execute.

• Object: Receptors of actions. Examples include files, programs, messages, records, terminals, printers, and user- or program-created structures. When a subject is the recipient of an action, such as electronic mail, then that subject is considered an object. Objects may be grouped by type. Object granularity may vary by object type and by environment. For example, database actions may be audited for the database as a whole or at the record level.

• Exception-Condition: Denotes which, if any, exception condition is raised on return.

• Resource-Usage: A list of quantitative elements in which each element gives the amount used of some resource (e.g., number of lines printed or displayed, number of records read or written, processor time, I/O units used, session elapsed time).

• Time-Stamp: Unique time-and-date stamp identifying when the action took place.

References

Barr, T. Invitation to Cryptology. Upper Saddle River, NJ: Prentice Hall, 2002.

Bishop, D. Cryptography with Java Applets. Sudbury, MA: Jones and Bartlett, 2003.

Bishop, M. Computer Security. Reading, MA: Addison-Wesley, 2005.

Blahut, U. Algebraic Codes for Data Transmission. Cambridge: Cambridge University Press, 2003.

Brassoud, D., and Wagon, S. Computational Number Theory. Emerville, CA: Key College, 2000.

Coutinho, S. The Mathematics of Ciphers. Natick, MA: A. K. Peters, 1999. [DF04] Dummit, D., and Foote, R. Abstract Algebra. Hoboken, NJ: John Wiley & Sons, 2004.

Doraswamy, H., and Harkins, D. IPSec. Upper Saddle River, NJ: Prentice Hall, 2003.

Durbin, J. Modern Algebra. Hoboken, NJ: John Wiley & Sons, 2005.

Enge, A. Elliptic Curves and Their Applications to Cryptography. Norwell, MA: Kluver Academic, 1999.

Forouzan, B. TCP/IP Protocol Suite. New York: McGraw-Hill, 2006.

Forouzan, B. Data Communication and Networking. New York: McGraw-Hill, 2007.

Frankkel, S. Demystifying the IPSec Puzzle. Norwood, MA: Artech House, 2001.

Kahn, D. The Codebreakers: The Story of Secret Writing. New York: Scribner, 1996.

Kaufman, C., Perlman, R., and Speciner, M. Network Security. Upper Saddle River, NJ: Prentice Hall, 2001.

Larson, R., Edwards, B., and Falvo, D. Elementary Linear Algebra. Boston: Houghton Mifflin, 2004.

Mao, W. Modern Cryptography. Upper Saddle River, NJ: Prentice Hall, 2004.

Menezes, A., Oorschot, P., and Vanstone, S. Handbook of Applied Cryptograpy. New York: CRC Press, 1997.

Pieprzyk, J., Hardjono, T., and Seberry, J. Fundamentals of Computer Security. Berlin: Springer, 2003.