Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

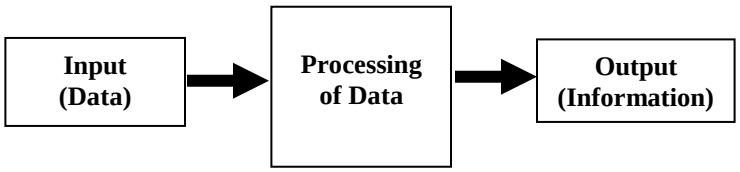
Data Security
First Semester
Lecture 1

Contents

Data Security Concepts

Introduction

Before studying data security, it is important to understand what the difference is between data and information. **Data** is a collection of raw details or data remaining in the form of either texts, symbols, descriptions, or mere observations of entities, events, or things with a potential to be analyzed and drawn inferences from (for example, patient data such as pressure data, diabetes ...etc.). **Information** is data collated to derive meaningful inferences according to its contextual requirement. Information is structured, processed, and presented with assigned meaning that improves the reliability of the data acquired (For example, patient information such as name, age ...etc.).



Also, there are two other important terms security and privacy. The security term is more comprehensive than privacy, as privacy can be part of security. **Security** refers to protective measures put in place to protect digital data from unauthenticated users, such as cyber criminals and hackers. **Privacy** is one's right to freedom from intrusion and prying eyes. Security is the state of personal freedom or being free from potential threats, whereas privacy refers to the state of being free from unwanted attention.

Data security is the sum of all measures taken to prevent loss of data. Loss can occur because of user error, defects in code, malicious acts, hardware failure, and acts of nature. Data/information security is applied to computing devices such as computers, sensors and smartphones, as well as computer networks such as private and public networks, including the whole Internet. Computer security definition is a techniques for ensuring that data stored in a computer cannot be read or compromised by any individuals without authorization. In addition, there is a difference between information security and cyber security. Information security can be physical (paper) while cyber security is concerned with protecting electronic information. The term cybersecurity is used instead of electronic security when dealing with networks.

We live in a world where "information wants to be free" and in which people are getting used to having access to whatever information they want anytime, anywhere and from a wider and wider range of computing devices. Unfortunately, in terms of the security and control of the resources to which computers permit access, this can prove quite a problem. Indeed, many users unfortunately often view security and control measures as inhibitors to effective computer use.

Security and Data Privacy Threats

The range of means by which the security and integrity of computing resources can be threatened is very broad, and encompasses:

- Operator error (for example a user inadvertently deleting the wrong file).
- Hardware or media failure (either as a result of wear-and-tear, old age or accidental damage).
- Theft or sabotage (of hardware and/or data or its media).
- Hackers (who obtain unauthorised online access via the Internet).
- Malware (any form of virus, and including "Trojan" e-mail attachments that users are encouraged to open).
- Power surges and/or outages (which are one of the most common means of hard disk corruption and hardware damage).
- Flood, fire, storm or other natural disasters.
- Fraud or embezzlement.
- Industrial espionage.
- Terrorism.

Why user data is not secure?

Most people question is why user data in computers are so insecure (where there is no 100% security)? The vast majority of hacking incidents occur because of one of the following pervasive problems:

1- Security techniques hinder users

Administrators often fail to implement security features in operating systems because doing so causes problems for users. Users also circumvent security-by choosing easy-to-use (easy to- guess) passwords like “123456,” never changing those passwords, disclosing those passwords to co-workers, or sharing user

accounts. The fact that strong security is an annoyance that requires extra learning on the part of everyone involved is the most common reason for security failures.

2- Corporate business interests

Vendors concentrate their efforts on adding features that make their software more useful, with little thought to security because security technologies are costly in terms of money, time and effort.

3- Viruses

Virus is any program that automatically replicates itself. When the Internet first took off, “e-mail Virus” scares propagated around the Net via e-mail. Computer security experts ignored them, knowing that a virus required an execution environment like a computer language in order to actually propagate. Viruses often perform some type of harmful activity on infected hosts, such as stealing hard disk space or CPU time, accessing private information, corrupting data, displaying political or humorous messages on the user's screen, spamming their contacts, logging their keystrokes, or even rendering the computer useless. However, the rapid spread of computer viruses (continuously developed) through storage devices or the Internet made 100% data security almost impossible.

4- Hackers threats

Hacking is quite simply the attempt to gain access to a computer system without authorization. A hacker is someone who seeks and exploits weaknesses in a computer system or computer network. Hackers may be motivated by a multitude of reasons, such as profit, protest, challenge, enjoyment, or to evaluate those weaknesses to assist in removing them. There is a huge competition between hackers and providers, hackers are trying to find loopholes in systems to penetrate data security and providers are trying to fill these gaps. Therefore, data security is a very important but very difficult matter.

H.W:

1- Is privacy part of security? Explain

2- What is the similarity between information and data?

3- Is there a difference between cyber security and electronic security?

4- Why do hackers penetrate user data or networks?

5- How can security measures hinder users? Give an example.

Data Security
4th Class

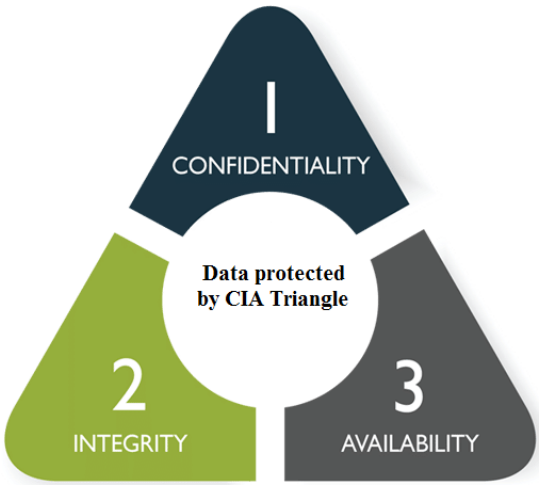
Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 2

Contents

Introduction

Any electronic application/platform that requires the protection of stored or transmitted data from change, modification, deletion...etc. To accomplish this, that application requires the inclusion of data protection requirements. There are three main elements to data security that all organizations/institutions should commit to: Confidentiality, Integrity, and Availability. These concepts are also referred to as the CIA triangle, functioning as a security model and framework for top-notch data security. In addition, there are a set of secondary requirements that support data security, as we will see in the next section.



Data Security Requirements

The main security requirements are as follows:

1. Confidentiality

This concept means hiding data with encryption to prevent illegal users from accessing it. It ensures that data is accessed only by legitimate users that have the proper secrets.

2. Integrity

It ensures that all data is reliable, accurate, and did not changed by attackers when it is stored or transmitted from the sender to the receiver, Integrity includes

- Data integrity: Assures that data are changed only in a specified and authorized manner
- System integrity: Assures that a system performs its operations in unimpaired manner.

3. Availability

This concept guarantees that data is readily and safely, accessible and available for ongoing business needs. Availability is a natural result of the other two concepts (Confidentiality and Integrity).

- Threats to Availability:
  - Availability can be affected by a number of events which break down into human and non-human influenced factors. These further breaks down to unintentional and intentional acts.
    - Examples of unintentional (non-directed) acts can be overwriting, in part or whole, of data, compromising of systems, or network infrastructure by organizational staff.
    - Intentional acts can be conventional warfare (bombs and air-strikes), information warfare denial of service (DoS) and distributed denial of service (DDoS).
    - Non-human factors include loss of availability due to fires, floods, earthquakes and storms.

The secondary security requirements are as follows:

1. Authentication

This requirement refers specifically to accurately identifying users before they have access to data. To be user authenticated in the system, it should prove his/her identity is authentic. Ways of performing authentication are:

- User ID and passwords: The system compares the given password with a stored password. If the two passwords match then the user is authentic.
- Swipe card: It has a magnetic strip embedded, which would already contain your details, so that no physical data entry takes place or just a PIN is entered.
- Digital certificate: An encrypted piece of data which contains information about its owner, creator, generation and expiration dates, and other data to uniquely identify a user.
- Key fob: Small electronic devices which generate a new random password synchronized to the main computer.
- Biometrics: Retinal scanners and fingerprint readers. Parts of the body are considered unique enough to allow authentication to computer systems based on their properties.

For a very secure environment, it is also possible to combine several of these options, such as by having fingerprint identification along with user ID and key fob.

2. Accountability (Non-Repudiation)

This concept ensures that no party can deny that it sent or received a data via encryption or digital signatures. It also cannot deny the authenticity of its signature

on a data. 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.

3. Access Control

This concept provides permissions and privileges to authorize users, employees and third parties to access institutions data in a manner that meets security, privacy and compliance requirements. The main purpose of access control is to ensure that access to resources (data) within an organization complies with the rules, policies and official regulations. The level of access depends on the role, attributes, parameters of the users ... etc.

Also there are other requirements that support data security: scalability, auditability, survivability ... etc.

Passwords and Appropriate user authentication

Physically protecting computer equipment and data against damage or loss is a large element of computer security. However, another large element is limiting access to all or part of a system or data store to authorized users only. In the broadest of terms, user authorization within any security system can be verified via one three means:

- Something known by the individual (a piece of information such as a password)
- Something possessed by the individual (a physical token such a credit, security or ID card), or
- A biometric characteristic of the individual (for example their signature, finger print, retinal scan or DNA).

For good security, two of the above measures should be employed for what is known as "two-factor security". For instance, to obtain money from a bank cash machine both a card and a PIN (personal identification number password) are required. Where computer security is concerned, one measure of user verification will almost always be a password given the relative technical ease with which this can be implemented.

Computer keyboards, mobile computers and dedicated input devices that include finger print readers are also becoming more common, and can be combined with password to achieve two-factor security. This in turn means that users must be educated to use strong passwords or in other words, to choose and use passwords in a manner that makes the password difficult to either fathom or otherwise obtain by an unauthorized party. To be classed as "strong", passwords,

- Should be at least six and preferably eight or more characters in length.
- Should be mixed case alphanumeric (a mix of apparently random upper and lower case letters and numbers is best).
- Should be changed regularly (at least every three months is a common rule).
- Should be known only to the user.
- Should not be obviously related to the user.
- Should be different for each application used
- Should not be based on data (such as a favourite place) listed publically on Facebook or another social networking site.

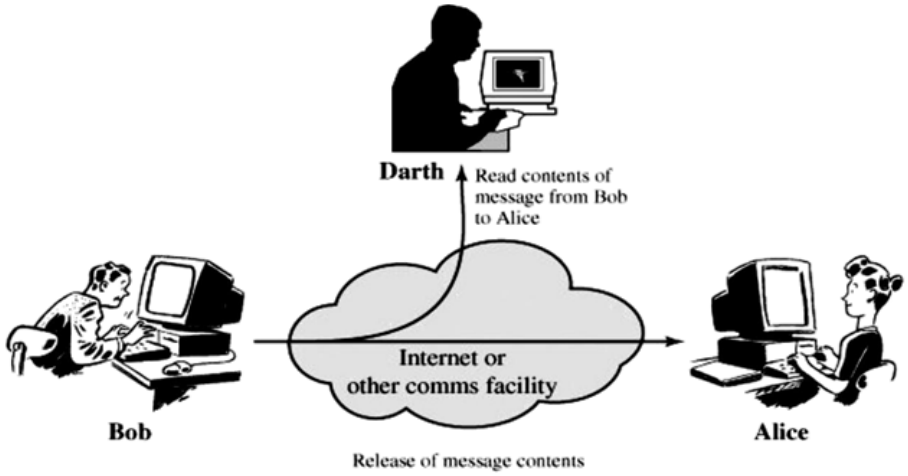
Security Attacks

Data breach attacks are categorized into two main categories, passive and active. A passive attack attempts to listen, learn or make use of information from the system but does not affect system data. An active attack attempts to modify system data or affect their operation.

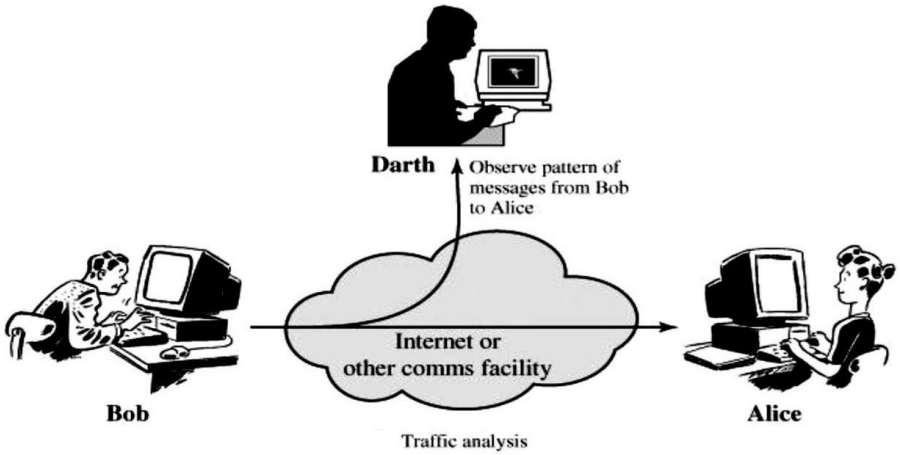
Passive Attacks

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

- The **release of message contents** is easily understood (in Figure, we assume that Bob is the sender, Alice is the receiver, and Darth is the hacker). 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. This type of attack simply tries to analyze the traffic, even if it is encrypted, to get the clear data. Suppose that we had a way of masking the contents of messages or other information traffic so that adversaries, 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 adversary might still be able to observe the pattern of these messages. The adversary 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.



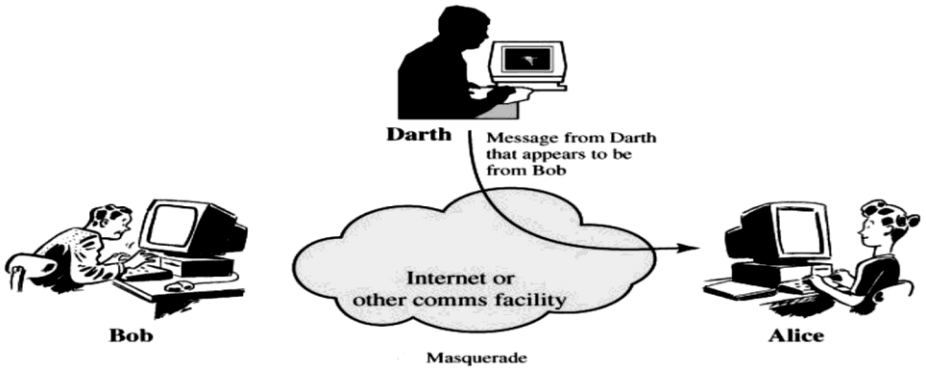
Passive attacks are very difficult to detect because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, if the

confidentiality requirement is properly implemented it can be a solution to prevent this type of attack.

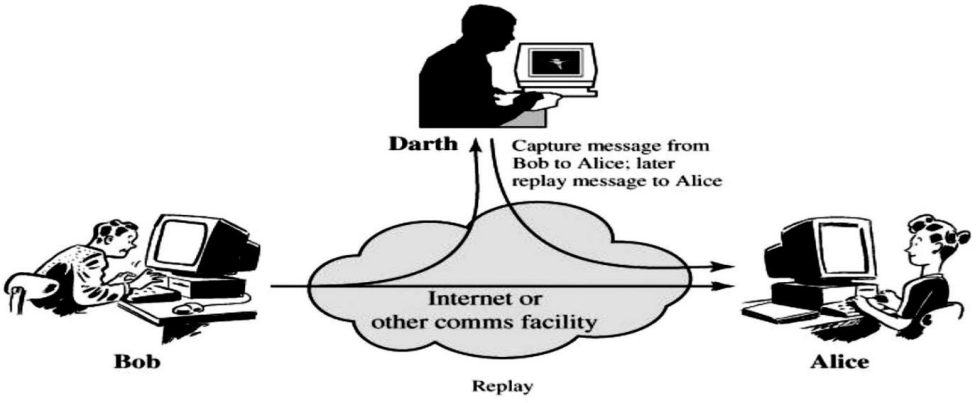
Active Attacks

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 data, and denial of service (DoS).

- **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.

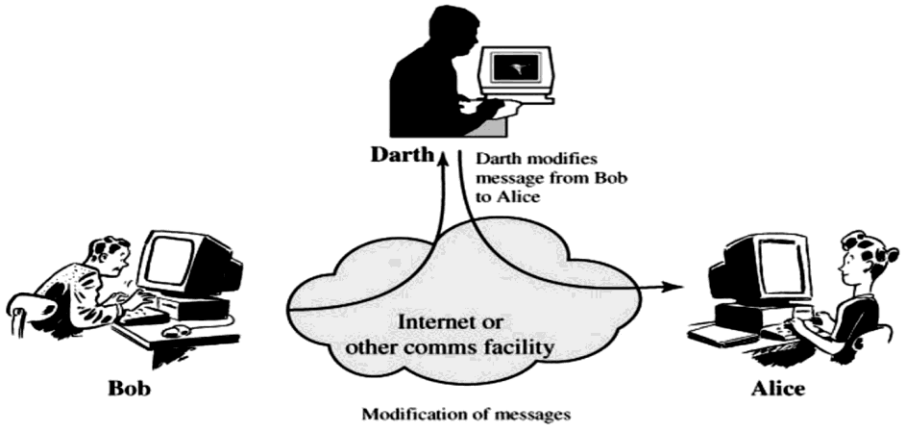


- **Replay** involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect.

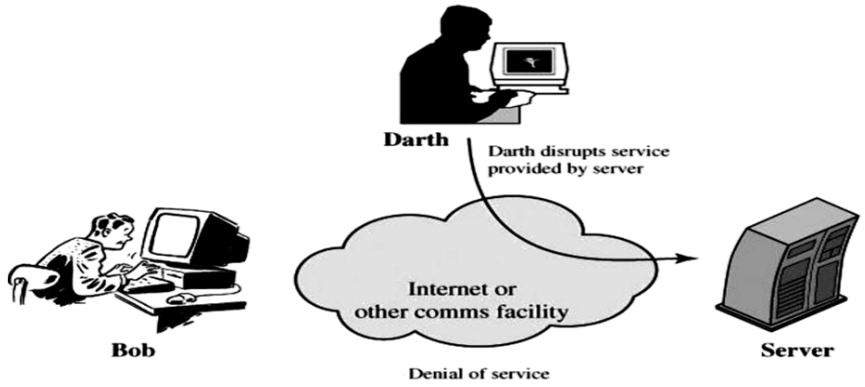


- **Modification of data**: simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (such as man-in-the-middle (MITM) attack). For instance, a message meaning "Allow John Smith to read confidential file

accounts" is modified to mean "Allow Fred Brown to read confidential file accounts."



- **Denial of service (DoS)**: It 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 (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. 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.

H.W:

1- Define:

- CIA triangle -Availability -Data security requirements - DoS

2- What is the difference between authentication and access control?

3- What is the difference between passive and active attacks?

4- How can we prevent passive attacks?

5- What attacks are more destructive, passive or active attack? And why?

Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 3

Contents

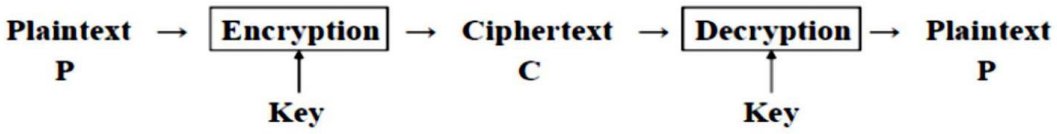
Basic Terminology

Suppose that sender wants to send a message to a receiver, and wants to be sure that no-one else can read the message. However, there is the possibility that someone else opens the message or hears the electronic communication. In cryptographic terminology, the message is called **plaintext** or **clear text**. Encoding the contents of the message in such a way that hides its contents from outsiders is called **encryption**. The encrypted message is called the **cipher text**. The process of retrieving the plaintext from the cipher text is called **decryption**. Encryption and decryption usually make use of a **key**, and the coding method is such that decryption can be performed only by knowing the proper key.

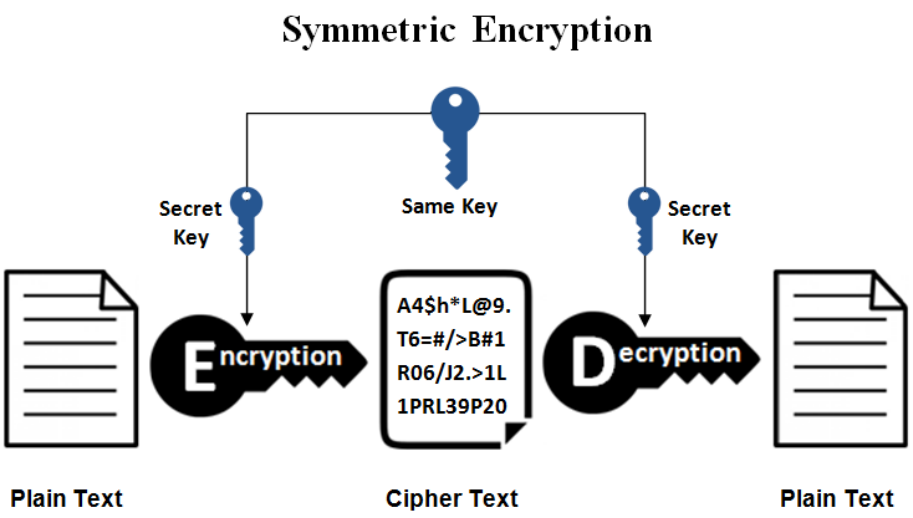
Cryptography is the art or science of keeping messages secret. **Cryptanalysis** is the art of breaking ciphers, i.e. retrieving the plaintext without knowing the proper key. Cryptography deals with all aspects of secure messaging, authentication, digital signatures, electronic money, and other applications. Cryptology is the branch of mathematics that studies the mathematical foundations of cryptographic methods.

Basic Cryptographic Algorithms

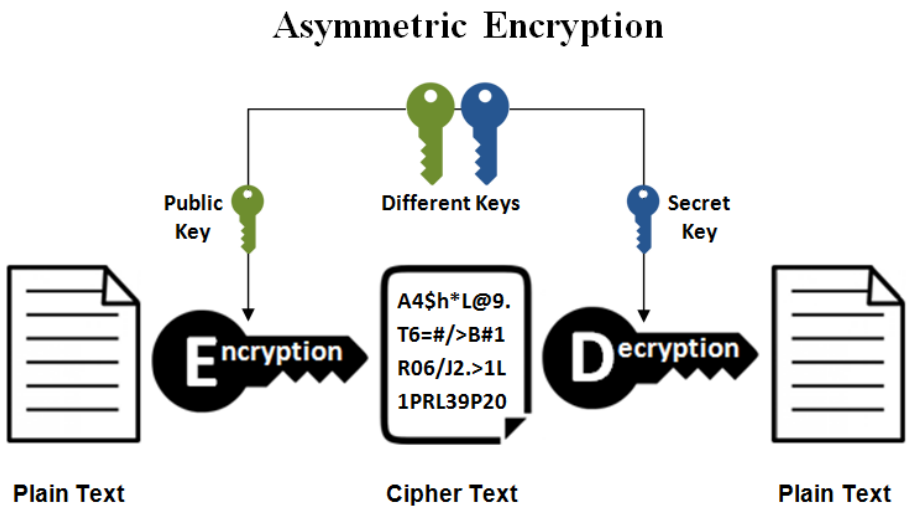
A method of encryption and decryption is called a cipher. Some cryptographic methods rely on the secrecy of the algorithms. All modern algorithms use a **key** to control encryption and decryption; a message can be decrypted only if the key matches the encryption key. The key used for decryption can be different from the encryption key, but for other algorithms they are the same.



There are two classes of key-based algorithms, **symmetric** (or secret-key) and **asymmetric** (or public-key) algorithms. The difference is that symmetric algorithms use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key), whereas asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be derived from the encryption key.



Symmetric algorithms can be divided into **stream ciphers** and **block ciphers**. Stream ciphers can encrypt a single bit of plaintext at a time, whereas block ciphers take a number of bits (typically 64 bits in modern ciphers), and encrypt them as a single unit.



Asymmetric ciphers (also called **public-key algorithms** or generally **public-key cryptography**) permit the encryption key to be public, allowing anyone to encrypt with the key, whereas only the proper recipient (who knows the decryption key) can decrypt the message. The encryption key is also called the **public key** and the decryption key the **private key** or **secret key**.

Modern cryptographic algorithms cannot really be executed by humans. Strong cryptographic algorithms are designed to be executed by computers or specialized

hardware devices. In most applications, cryptography is done in computer software, and numerous cryptographic software packages are available. Generally, symmetric algorithms are much faster to execute on a computer than asymmetric ones. In practice they are often used together, so that a public-key algorithm is used to encrypt a randomly generated encryption key, and the random key is used to encrypt the actual message using a symmetric algorithm.

Strength of Cryptographic Algorithms

Good cryptographic systems should always be designed so that they are as difficult to break as possible. It is possible to build systems that cannot be broken in practice (though this cannot usually be proved). This does not significantly increase system implementation effort; however, some care and expertise is required. There is no excuse for a system designer to leave the system breakable. Any mechanisms that can be used to circumvent security must be made explicit, documented, and brought into the attention of the end users. In theory, any cryptographic method with a key can be broken by trying all possible keys in sequence. If using brute force to try all keys is the only option, the required computing power increases exponentially with the length of the key.

- A 32 bit key takes 2^32 (about $10^{\wedge 9}$) steps. This is something any amateur can do on his/her home computer.
- A system with 56 bit keys takes $2^{\wedge 56}$ steps - this kind of computing power is available in most universities and even smallish companies.
- A system with 56 bit keys takes a substantial effort, but is quite easily breakable with special hardware.

Keys with 64 bits are probably breakable now by major governments, and will be within reach of organized criminals, major companies, and lesser governments in a few years.

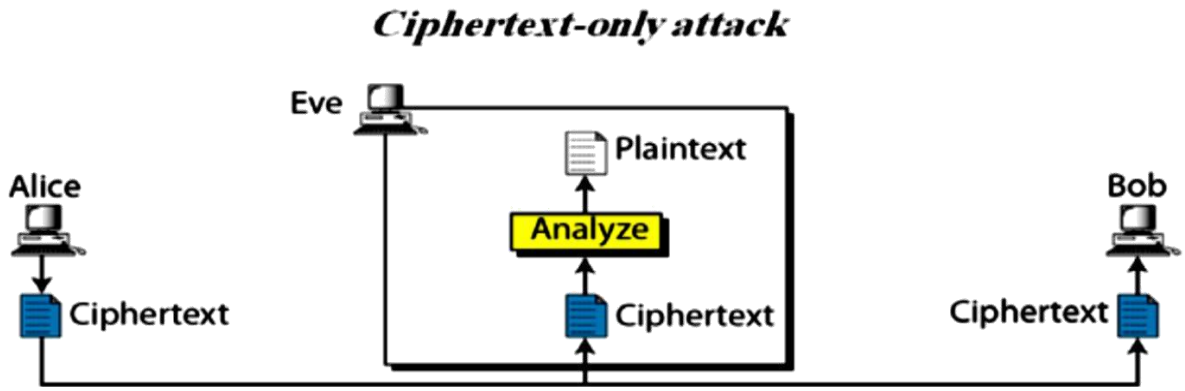
- Keys with 80 bits may become breakable in future.
- Keys with 128 bits will probably remain unbreakable by brute force for the foreseeable future. However, key length is not the only relevant issue. Many ciphers can be broken without trying all possible keys.
- Designing your own ciphers may be fun, but it is not recommended in real applications unless you are a true expert and know exactly what you are doing.
- To give some idea of the complexity, for the RSA cryptosystem, a 256 bit modulus is easily factored by ordinary people. 384 bit keys can be broken by university research groups or companies. 512 bits is within reach of major

governments. Keys with 768 bits are probably not secure in the long term. Keys with 1024 bits and more should be safe for now unless major algorithmic advances are made in factoring; keys of 2048 bits are considered by many to be secure for decades.

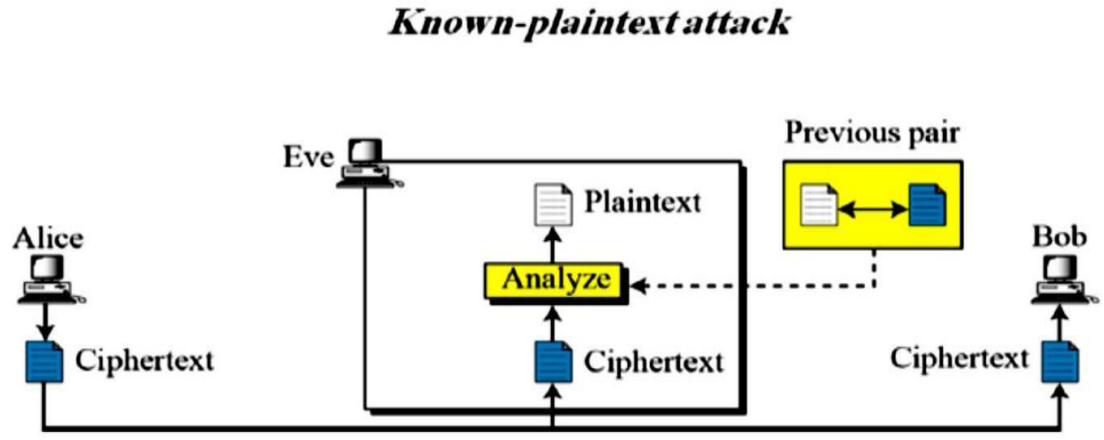
Cryptanalysis and Attacks on Cryptosystems

Cryptanalysis is the art of deciphering encrypted communications without knowing the proper keys. There are many cryptanalytic techniques.

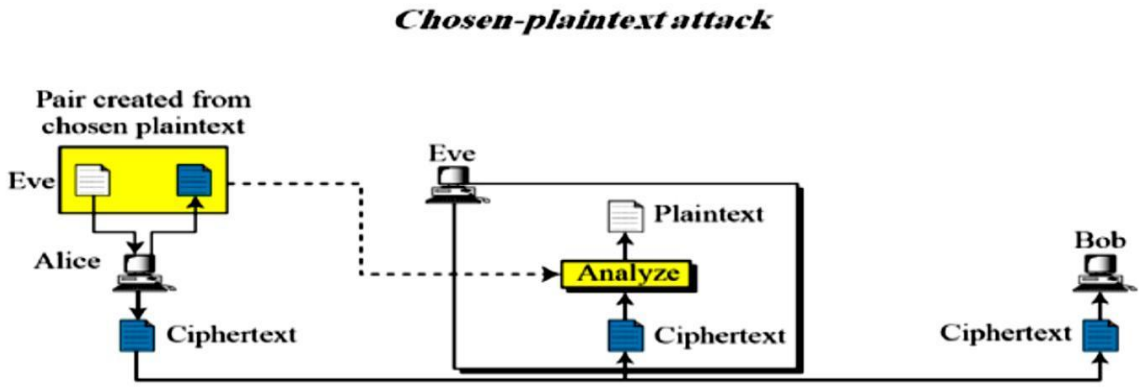
- **Ciphertext-only attack:** This is the situation where the attacker does not know anything about the contents of the message, and must work from ciphertext only. In practice, it is quite often possible to make guesses about the plaintext, as many types of messages have fixed format headers. Even ordinary letters and documents begin in a very predictable way. It may also be possible to guess that some ciphertext block contains a common word.



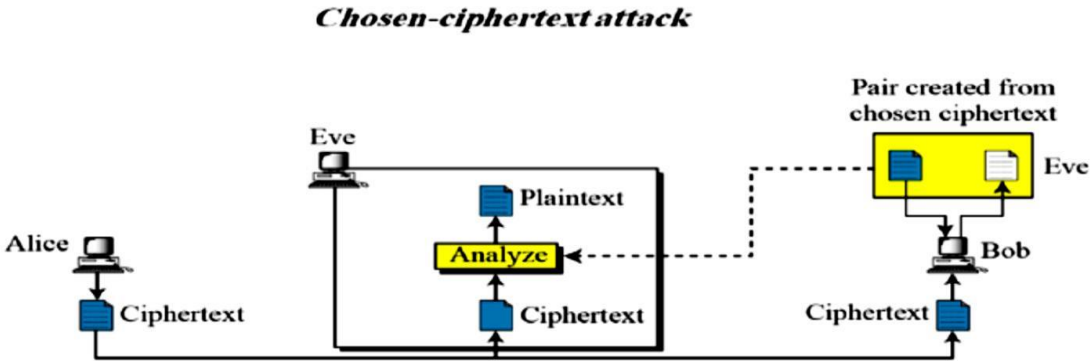
- **Known-plaintext attack:** The attacker knows or can guess the plaintext for some parts of the ciphertext. The task is to decrypt the rest of the ciphertext blocks using this information. This may be done by determining the key used to encrypt the data, or via some shortcut.



- **Chosen-plaintext attack:** The attacker is able to have any text he/she likes encrypted with the unknown key. The task is to determine the key used for encryption. Some encryption methods, particularly RSA, are extremely vulnerable to chosen-plaintext attacks. When such algorithms are used, extreme care must be taken to design the entire system so that an attacker can never have chosen plaintext encrypted.



- **Chosen Ciphertext Attacks** (select ciphertext and obtain plaintext to attack cipher): Attacker obtains the decryption of any ciphertext of its choice (under the key being attacked).



- **Man-in-the-middle attack:** This attack is relevant for cryptographic communication and key exchange protocols. The idea is that when two parties are exchanging keys for secure communications, an adversary puts himself/hereself between the parties on the communication line. The adversary then performs a separate key exchange with each party. The parties will end up using a different key, each of which is known to the adversary. The adversary will then decrypt any communications with the proper key, and encrypt them with the other key for sending to the other

party. The parties will think that they are communicating securely, but in fact the adversary is hearing everything. One way to prevent man-in-the-middle attacks is that both sides compute a cryptographic hash function of the key exchange (or at least the encryption keys), sign it using a digital signature algorithm, and send the signature to the other side. The recipient then verifies that the signature came from the desired other party, and that the hash in the signature matches that computed locally.

- **Timing Attack:** This very recent attack is based on repeatedly measuring the exact execution times of modular exponentiation operations.

Cryptographic Random Number Generators

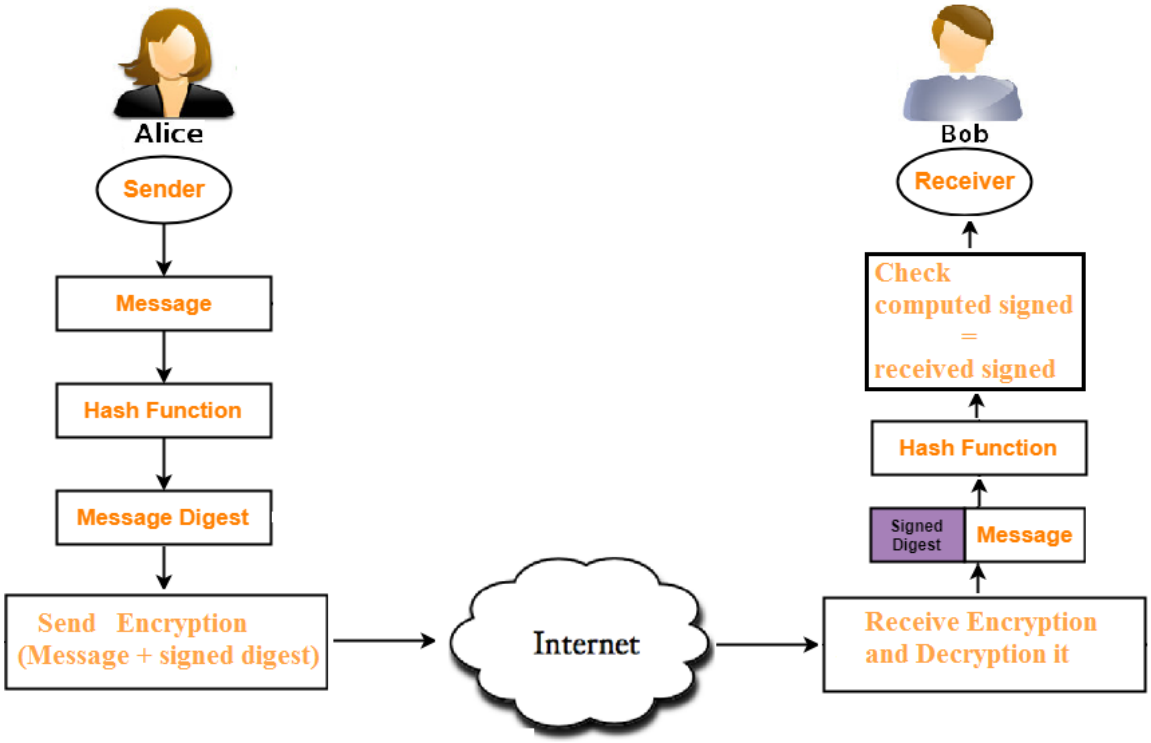
These generators generate random numbers for use in cryptographic applications, such as for keys. Conventional random number generators available in most programming languages or programming environments are not suitable for use in cryptographic applications (they are designed for statistical randomness, not to resist prediction by cryptanalysts).

- In the optimal case, random numbers are based on true physical sources of randomness that cannot be predicted.
- When true physical randomness is not available, pseudorandom numbers must be used. This situation is undesirable, but often arises on general purpose computers.
  - Cryptographic pseudorandom generators typically have a large pool ("seed value") containing randomness. Bits are returned from this pool by taking data from the pool, optionally running the data through a cryptographic hash function to avoid revealing the contents of the pool.
  - Even though cryptographically strong random number generators are not very difficult to build if designed properly, they are often overlooked. The importance of the random number generator must thus be emphasized - if done badly; it will easily become the weakest point of the system.

Digital Signatures

Some public-key algorithms (such as ECC and RSA) can be used to generate digital signatures. A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or documents. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, that the sender cannot deny having sent the message (authentication and non-

repudiation), and that the message was not altered in transit (integrity). Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.



Explanation

Digital signatures are often used to implement electronic signatures, a broader term that refers to any electronic data that carries the intent of a signature, but not all electronic signatures use digital signatures. In some countries, electronic signatures have legal significance.

Digital signatures employ asymmetric cryptography. In many instances they provide a layer of validation and security to messages sent through a non-secure channel: Properly implemented, a digital signature gives the receiver reason to believe the message was sent by the claimed sender. Digital seals and signatures are equivalent to handwritten signatures and stamped seals. Digital signatures are equivalent to traditional handwritten signatures in many respects, but properly implemented digital signatures are more difficult to forge than the handwritten type.

Applications of digital signatures

As organizations move away from paper documents with ink signatures or authenticity stamps, digital signatures can provide added assurances of the evidence to

provenance, identity, and status of an electronic document as well as acknowledging informed consent and approval by a signatory. Universities including Penn State, University of Chicago, and Stanford are publishing electronic student transcripts with digital signatures. In addition, there are many applications that use electronic signatures such as e-health, e-government, e-banking, e-military applications ...etc. Below are some common reasons for applying a digital signature to communications:

Authentication

Although messages may often include information about the entity sending a message, that information may not be accurate. Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user. The importance of high confidence in sender authenticity is especially obvious in a financial context. For example, suppose a bank's branch office sends instructions to the central office requesting a change in the balance of an account. If the central office is not convinced that such a message is truly sent from an authorized source, acting on such a request could be a grave mistake.

Integrity

In many scenarios, the sender and receiver of a message may have a need for confidence that the message has not been altered during transmission. Although encryption hides the contents of a message, it may be possible to change an encrypted message without understanding it. However, if a message is digitally signed, any change in the message after signature invalidates the signature. Furthermore, there is no efficient way to modify a message and its signature to produce a new message with a valid signature, because this is still considered to be computationally infeasible by most cryptographic hash functions.

Non-repudiation

Non-repudiation, or more specifically non-repudiation of origin, is an important aspect of digital signatures. By this property, an entity that has signed some information cannot at a later time deny having signed it. Similarly, access to the public key only does not enable a fraudulent party to fake a valid signature.

Introduction of Cryptography

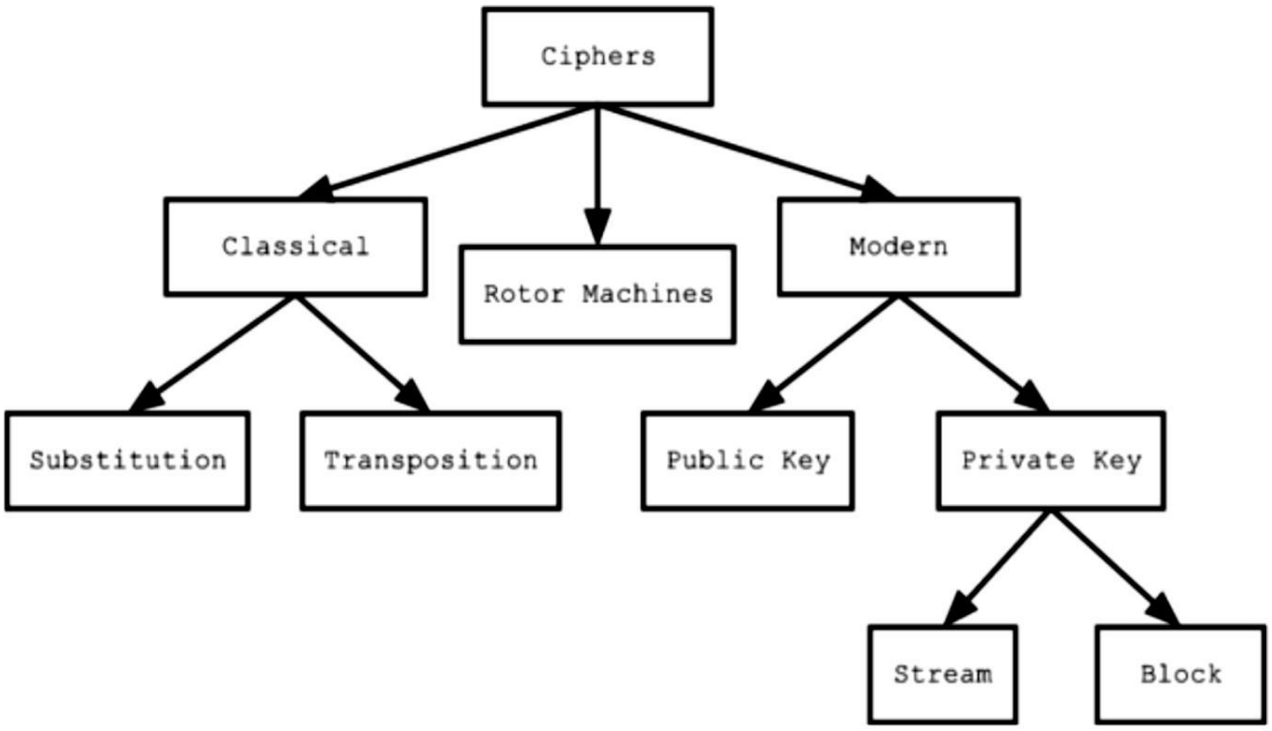
Computers are now fond in every layer society, and information is being communicated and processed automatically on a large scale. Such as medical and financial files, tele-shopping, and global computer networks. In all these cases there is a growing need for the protection of information to safeguard economic interests, to

prevent fraud and to ensure privacy. The term Cryptography is originally derived from the two greek words “kryptos" and “graph", meaning hidden and writing. This is an accurate representation of the meaning of the word, as cryptography is the art of ensuring that messages (writing) are kept secure (hidden) from those recipients to whom the messages are not addressed.

Cryptography is the science and study of methods of protecting data in computer and communication systems from unauthorized disclosure and modification. The Cryptographic systems are classified into two cryptosystems, private-key cryptosystem and public-key cryptosystem. Both are based on complex mathematical algorithms and are controlled by keys. The advances in cryptography have boosted its use in recent decades, opening up an amazing array of applications. It can be used to authenticate computer users, ensure the integrity and confidentiality of electronic communications, and keep sensitive information safely stored. From a secretive military technology, cryptography has emerged a key technology for all participants in the information society concerned about information security.

Types of Cryptography

There are a variety of different types of encryption. Algorithms used earlier in the history of cryptography are substantially different from modern methods, and modern ciphers can be classified according to how they operate and whether they use one or two keys.



H.W:

1- Define

- Plaintext - Clear text - Cryptanalysis - Timing attack -Decryption -Key

2- What are the security requirements provided by an electronic signature? Explain one of them.

3- What is the difference between secret key algorithms and public key algorithms?

4- Give an example of symmetric encryption algorithms.

5- What is the difference between Asymmetric algorithms and public key algorithms?

Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 4

Contents

Classical Encryption

This type of encryption is divided into three main types of encryption which are Monoalphabetic, Polyalphabetic and Polygraphic.

Monoalphabetic Ciphers

A monoalphabetic cipher uses the same substitution across the entire message. For instance, if user know that the letter A is enciphered as the letter K, this will hold true for the entire message. These types of messages can be cracked by using frequency analysis, educated guesses or trial and error. There are many algorithms for Monoalphabetic ciphers such as

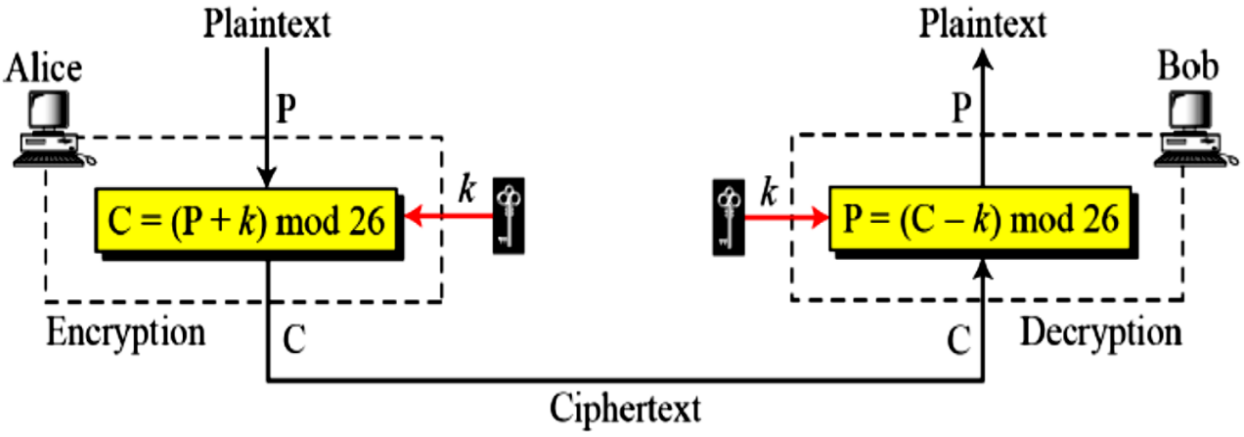
- **Caesar Cipher**
- **Atbash Cipher**
- **Keyword Cipher**
- **Polybius Square**

Caesar Cipher (Additive Cipher or shift cipher)

A Caesar cipher is one of the simplest encryption methods. It is a Substitution Cipher that involves replacing each letter of the secret message with a different letter of the alphabet which is a fixed number of positions further in the alphabet. Because each letter in the message has a direct translation to another letter, frequency analysis can be used to decipher the message. For instance, the letter E is the most commonly used letter in the English language. Thus, if the most common letter in a secret message is K, it is likely that K represents E. Additionally, common word endings such as ING, LY, and ES also give clues. A brute-force attack of trying all 25 possible combinations would also work to decipher the message.

Substitution equation: Cipher (C) = (Plaintext (P) + K) mod 26

Plaintext (p) →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext (C) →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



Example1: Find cipher text in Caser method using k=3

Plaintext: "**Data Security**"
Substitution equation: C = P + K mod 26

Sol: **D , 3+3 mod 26 = 6 = G**
a , 0+3 mod 26 = 3 = d
t , 19+3 mod 26 = 22 = w
a , 0+3 mod 26 = 3 = d

S , 18+3 mod 26 = 21 = V
e , 4+3 mod 26 = 7 = h
c , 2+3 mod 26 = 5 = f
u , 20+3 mod 26 = 23 = x
r , 17+3 mod 26 = 20 = u
i , 8+3 mod 26 = 11 = l
t , 19+3 mod 26 = 22 = w
y , 24+3 mod 26 = 1 = b

Ciphertext: "**Gdwd Vhfxulwb**"

Example2: Each letter in the plaintext message has been shifted 3 letters down in the alphabet.

Plaintext: "**The data is encrypted"**
Ciphertext: "**Wkh gdwd lv hqfubswhg**"

Example3: Each letter in the plaintext message has been shifted 3 letters down in the alphabet.

Plaintext: "**Authentication is important"**
Ciphertext: "**Dxwkhqwlfdwlrq lv lpsruwdqw**"

Atbash Cipher

The Atbash cipher is a very specific case of a substitution cipher where the letters of the alphabet are reversed. In other words, all letters of As are replaced with letters of Zs, all letters of Bs are replaced with letters of Ys, and so on. Because reversing the alphabet twice will get you actual alphabet, user can encipher and decipher a data using the exact same algorithm.

First 13 letters	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
Last 13 letters	**Z**	**Y**	**X**	**W**	**V**	**U**	**T**	**S**	**R**	**Q**	**P**	**O**	**N**

Example1: Find cipher text in atbash algorithm

Plaintext: "**Data Security**"
Ciphertext: "**Wzgz Hvxfirgb**"

Example2:

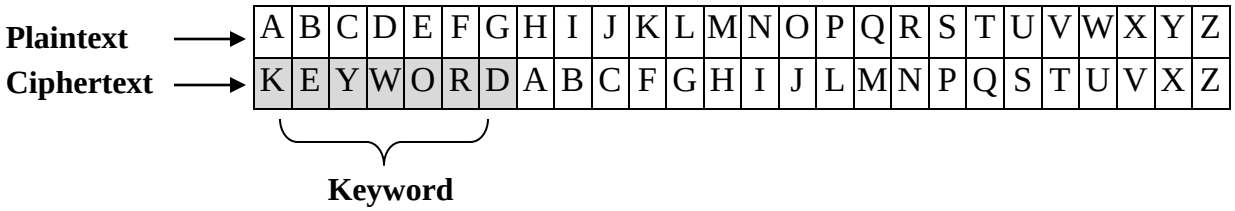
Plaintext: "**The data is encrypted"**
Ciphertext: "**Gsv wzgz rh vmxibkgvw**"

Example3:

Plaintext: "**Authentication is important"**
Ciphertext: "**Zfgsvmgrxzgrlm rh rnkligzmg**"

Keyword Cipher

The Keyword cipher is identical to the Caesar Cipher with the exception that the substitution alphabet used can be represented with a keyword. To create a substitution alphabet from a keyword, you first write down the alphabet. Below this you write down the keyword (omitting duplicate letters) followed by the remaining unused letters of the alphabet.



To encipher a plaintext message, you convert all letters from the top row to their corresponding letter on the bottom row (A to K, B to E, etc). These types of simple substitution ciphers can be easily cracked by using frequency analysis and some educated guessing.

Example1: Find cipher text in Keyword algorithm

Plaintext: "**Data Security**"
Ciphertext: "**wkqk poysnbqx**"

Example2:

Plaintext: "**The data is encrypted"**
Ciphertext: "**qao wkqk bp oiynxlqow**"

Example3:

Plaintext: "**Authentication is important"**
Ciphertext: "**ksqaoiqbykqbji bp bhljnqkiq**"

Polybius Square

A Polybius Square is a table that allows someone to translate letters into numbers. To give a small level of encryption, this table can be randomized and shared with the recipient. In order to fit the 26 letters of the alphabet into the 25 spots created by the table, the letters i and j are usually combined. To encipher a message user replace each letter with the row and column in which it appears. For instance, D would be replaced with 14. To decipher a message user find the letter that intersects the specified row and column.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Example1: Find cipher text in Keyword algorithm

Plaintext: "**Data Security**"
Ciphertext: "**14 11 44 11 43 15 13 45 42 24 44 54**"

Example2:

Plaintext: "**The data is encrypted"**
Ciphertext: "**44 23 15 14 11 44 11 24 43 15 33 13 42 54 35 44 15 14**"

Example3:

Plaintext: "**Authentication is important"**
Ciphertext: "**11 45 44 23 15 33 44 24 13 11 44 24 34 33 24 43 24 32 35 34 42 44 11 33 44**"

H.W:

1- In Caesar Cipher, find ciphertexts from plaintexts:
"Bob sends code to Alice" using k=3
"Data security is important" using k=4
"Privacy is part form security" using k=5

2- In Caesar Cipher, find deciphertexts from plaintexts:
"Lqirupdwlrq lv sduw ri wkh gdwd" using k=3
"wmkrexyvi eglmiziw mrxikvexmsr" using k=4
"Htsknijsynfqnyd rjfsx htshjfqrjsy" using k=5

3- In Atbash Cipher, find ciphertexts from plaintexts:
"Bob sends code to Alice"
"Data security is important"
"Privacy is part form security"

4- In Atbash Cipher, find deciphertexts from plaintexts:
"Rmulinzgrlm rh kzig lu gsv wzgz"
"hrtmzgfiv zxsrvevh rmgvtizgrlm"
"Xlmurwvmgrzorgb nvzmh xlmxvzonvmg"

5- In Keyword Cipher, find ciphertexts from plaintexts:
"Bob sends code to Alice" using keyword= "Privacy"
"Data security is important" using keyword= "keyword"
"Privacy is part form security" using keyword= "security"

6- In Keyword Cipher, find deciphertexts from plaintexts:
"djcknhpqdkj do lpnq kc qba vpqp" using keyword= "Privacy"
"pbdikqsno kyabotop biqodnkqbji" using keyword= "keyword"
"cjhiaurhoasfaox grshn cjhcrsfgrho" using keyword= "security"

7- In Polybius Square Cipher, find ciphertexts from plaintexts:
"Bob sends code to Alice"
"Data security is important"
"Privacy is part form security"

8- In Polybius Square Cipher, find deciphertexts from plaintexts:
"24 33 21 34 42 32 11 44 24 34 33 24 43 35 11 42 44 34 21 44 23 15 14 11 44 11"
"43 24 22 33 11 44 45 42 15 11 13 23 24 15 51 15 43 24 33 44 15 22 42 11 44 24 34 33"
"13 34 33 21 24 14 15 33 44 24 11 31 24 44 54 32 15 11 33 43 13 34 33 13 15 11 31 32 15 33 44"

Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 5

Contents

Polyalphabetic Ciphers

In a polyalphabetic cipher, the substitution may change throughout the message. In other words, the letter “A” may be encoded as the letter “K” for part of the message, but later on it might be encoded as the letter “W”.

- **Vigenère Cipher**
- **Beaufort Cipher**
- **Autokey Cipher**
- **Running Key Cipher**

Vigenère Cipher

In a Caesar Cipher, each letter of the alphabet is shifted along some number of places; for instance, in a Caesar cipher of shift 3, “A” would become “D”, “B” would become “E” and so on. The Vigenere cipher consists of using several Caesar ciphers in sequence with different shift values.

To encipher, a table of alphabets can be used, termed a tabula recta, Vigenère square, or Vigenère table. It consists of the alphabet written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar ciphers. At different points in the encryption process, the cipher uses a different alphabet from one of the rows. The alphabet used at each point depends on a repeating keyword.

Example1: Find cipher text in Vigenère method using keyword “**SECRET**”.
Plaintext: “**Data security**”

The user sending the message chooses a keyword and repeats it until it matches the length of the plaintext (Number of plaintext symbols). “**Data security**” consists of 12 symbols.

Keyword: "**SECRET**" it will be like “**SECRETSECRET**”

Keyword Letters

Plaintext Letters

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Each letter is encoded by finding the intersection in the grid between the plaintext letter and keyword letter. For instance, the first letter of the plaintext "D" is enciphered using the alphabet in row "S", which is the first letter of the key. This is done by looking at the letter in row "S" and column "D" of the Vigenere square, namely "V". Similarly, for the second letter of the plaintext, the second letter of the key is used; the letter at row "E" and column "A" is "E". The rest of the plaintext is enciphered in a similar fashion:

Plaintext: "**Data security**"
Key: "**SECRETSECRET**"
Ciphertext: "**Vevr wxuytzxr**"

Decryption is performed by finding the position of the ciphertext letter in a row of the table, and then taking the label of the column in which it appears as the plaintext. For instance, in row "S", the ciphertext "V" appears in column "D", which taken as the first plaintext letter. The second letter is decrypted by looking up "E" in row "E" of the table; it appears in column "A", which is taken as the plaintext letter.

Example2: Find cipher text in Vigenère method using keyword "**PRIVICY**".

Plaintext: "**The data is encrypted**"

Key: "**PRIVICYPRIVICYPRIV**"

Ciphertext: "**Iym yavy xj mictwekmy**"

Example3: Find cipher text in Vigenère method using keyword "**PROTECT**".

Plaintext: "**Authentication is important**"

Key: "**PROTECTPROTECTPROTECTPROT**"

Ciphertext: "**Plhaipmxtommqg xj wftqkirbm**"

Beaufort Cipher

A Beaufort cipher uses the same alphabet table as the Vigenère cipher, but with a different algorithm. To encode a letter user find the letter in the top row. Then trace down until user find the key letter. Then trace over to the left most column to find the enciphered letter. To decipher a letter, you find the letter in the left column, trace over to the key letter and then trace up to find the deciphered letter. Some people find this easier to do than finding the intersection of a row and column.

Plaintext Letters

Ciphertext Letters

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Example1: Find cipher text in Beaufort method using keyword "**SECRET**".

Plaintext: "**Data Security**"
Key: "**SECRETSECRET**"
Ciphertext: "**Pejr Mpqkljlv**"

Example2: Find cipher text in Beaufort method using keyword "**PRIVICY**".

Plaintext: "**The data is encrypted**"
Key: "**PRIVICYPRIVICYPRIV**"
Ciphertext: "**Wke sijy hz eiglaayes**"

Example3: Find cipher text in Beaufort method using keyword "**PROTECT**".

Plaintext: "**Authentication is important**"
Key: "**PROTECTPROTECTPROTECTPROT**"
Ciphertext: "**Pxvmapahpoawog hz ghpocwrba**"

Autokey Cipher

An Autokey cipher is identical to the Vigenère cipher with the exception that instead of creating a keyword by repeating one word over and over, the keyword is constructed by appending the keyword to the beginning of the actual plaintext message.

For instance, if your plaintext message was: "**Server provides a strong security**"
And your keyword was "**Hide**", then your actual keyword would be:

"**HideServerprovidesastrongsecu**"

Enciphering and deciphering the message is performed using the exact same method as the Vigenère Cipher.

Example1: Find cipher text in Autokey method using keyword "**SECRET**".

Plaintext: "**Data Security**"
Key: "**SECRETDATASE**"
Ciphertext: "**VEVR WXFUKILC**"

Example2: Find cipher text in Autokey method using keyword "**PRIVICY**".

Plaintext: "**The data is encrypted**"
Key: "**PRIVICYTHEDATAISEN**"
Ciphertext: "**IYM YIVY BZ IQCKYXLIQ**"

Example3: Find cipher text in Autokey method using keyword "**PROTECT**".

Plaintext: "**Authentication is important**"
Key: "**PROTECTAUTHENTICATIONISIM**"

Ciphertext: "**PLHAIPMIWTAMBG QU IFXCEBSVF**"

Running Key Cipher

In a Running Key cipher, the keyword is the text of a predetermined book or passage. For instance, if they chose book was "A Tale of Two Cities" by Charles Dickens, then the keyword would be "It was the best of times, it was the worst of times".

Enciphering and deciphering the message is performed using the exact same method as the Vigenère Cipher. If the predetermined passage is a string of random letters that is only used once and then discarded, this is similar to a One-time Pad.

Example1: Find cipher text in Running Key method
Using sentence "**This is a good technique**".
Plaintext: "**Data Security**"
Key: "**Thisisagoodt**"

Ciphertext: "**whbsawcafwwr**"

Example2: Find cipher text in Running Key method
Using sentence "**It was the best of times**".
Plaintext: "**The data is encrypted**"
Key: "**Itwasthebestoftime**"

Ciphertext: "**baadsmhmtifvfdibqh**"

Example3: Find cipher text in Running Key method
Using sentence "**Their suggest were wrong and unclear**".
Plaintext: "**Authentication is important**"
Key: "**Theirsuggestwerewrongandu**"

Ciphertext: "**tbxpvfnoielbkrzweddbxtnqn**"

H.W:

1- In **Vigenère** Cipher, find ciphertexts from plaintexts:
"Bob sends code to Alice" using keyword "Info".
"Data security is important" using keyword "strong".
"Privacy is part form security" using keyword "secure".

2- In **Vigenère** Cipher, find deciphertexts from ciphertexts:
"Buj vippsz vx zwbpwvt kcz n qcwcmctr" using keyword "Info".
"Agkstxamp wf zg iicgkum uogg xkfa pnsgxs" using keyword "strong".
"Kieoimlc rfrck ep cdtgvvuex jsny zr Werjcmuevcfrk" using keyword "secure".

3- In **Beaufort** Cipher, find ciphertexts from plaintexts:
"Server provides services to clients" using keyword "NETWORK".
"DoS attacks disrupt network services" using keyword "PROTOCOL".
"Each network has different services" using keyword "KEY".

4- In **Beaufort** Cipher, find decryptions from ciphertexts:
"Mjqpglml jq hnp qk ybr oplidfd mo xppynnn" using keyword "PRIVATE".
"Iklxjr mosi ug ukuk kucwdpudl" using keyword "POLICY".
"Mejawlh wjfq qv xoexnp jczruwevza fz xwiuqay" using keyword "NEWKEY".

5- In **Autokey** Cipher, find ciphertexts from plaintexts:
"Availability is different from integrity" using keyword "SAFE".
"Attacks are a serious threat" using keyword "HIDE".
"Security techniques is a critical measure" using keyword "ALERT".

6- In **Autokey** Cipher, find decryptions from ciphertexts:
"LVP VXMHGKXK CJKRXGZ HVY WLYILWZZEL" using keyword "SOLVE".
"ICOEGFO OF LVLNWUMSY JISZ XSTFKTGKFL" using keyword "QUIRY".
"TLJTVFEGTDNP IH H ZZGAE JQIHFEM" using keyword "RULE".

7- In **Running Key** Cipher, find ciphertexts from plaintexts:
"Privacy is security" using sentence "The attacker is trying to break through"
"Information is data" using sentence "The data is precisely distributed across systems"
"Record is part of database" using sentence "Banks is a valuable target for hackers".

8- In **Running Key** Cipher, find deciphertexts from ciphertexts:
"LLRUIR JIIDJ LONS" using sentence "There are various security methods"
"UPKVRV PMUEUZXZ RDHF" using sentence "Secrecy is a method of hiding data"
"LNYOZ VU OUWE" using sentence "Data includes rows and columns"

Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 6

Contents

Polygraphic Ciphers

Instead of substituting one letter for another letter, a polygraphic cipher performs substitutions with two or more groups of letters. This has the advantage of masking the frequency distribution of letters, which makes frequency analysis attacks much more difficult.

- **Playfair Cipher**
- **Bifid Cipher**
- **Trifid Cipher**
- **Four-square cipher**
- **Hill Cipher**

Playfair Cipher

The Playfair cipher encrypts pairs of letters (digraphs), instead of single letters. This is significantly harder to break since the frequency analysis used for simple substitution ciphers is considerably more difficult. Memorization of the "**Hello World**" and 4 simple rules is all that is required to create the 5 by 5 table and use the cipher.

H	**E**	**L**	**O**	**W**
R	**D**	**A**	**B**	**C**
F	**G**	**I**	**J**	**K**
M	**N**	**P**	**S**	**T**
U	**V**	**X**	**Y**	**Z**

The Playfair cipher uses a 5 by 5 table containing a key word or phrase. To generate the table, one would first fill in the spaces of the table with the letters of the keyword (dropping any duplicate letters), then fill the remaining spaces with the rest of the letters of the alphabet in order (to reduce the alphabet to fit user can either omit "Q" or replace "J" with "I").

To encrypt a message, one would break the message into groups of 2 letters. If there is a dangling letter at the end, we add an X. For example. "Hide the gold" becomes "HI DE TH EG OL DX". We now take each group and find them out on the table. Noticing the location of the two letters in the table, we apply the following rules, in order.

1. If both letters are the same, add an X between them. Encrypt the new pair, re-pair the remining letters and continue.

2. If both letters are in the same **column**, take the letter below each one (going back to the top if at the bottom).
3. If both letters are in the same **row**, take the letter to the right of each one (going back to the left if at the farthest right).
4. If the letters are on different rows and columns, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. The order is important - the first letter of the pair should be replaced first.

Using these rules, the result of the encryption of "hide the gold" with the key of "hello world" would be "LF GD MW DN WO AV". To decipher, ignore rule 1. In rules 2 and 3 shift up and left instead of down and right. Rule 4 remains the same. Once you are done, drop any extra Xs that do not make sense in the final message and locate any missing Qs or any Is that should be Js.

Example1: Find cipher text in Playfair method.

Keyword: "**SECRET**"
Plaintext: "**Data security**"

Formatted Plaintext: "DA TA SE CU RI TY"

S	**E**	**C**	**R**	**T**
A	**B**	**D**	**F**	**G**
H	**I**	**J**	**K**	**L**
M	**N**	**O**	**P**	**U**
V	**W**	**X**	**Y**	**Z**

Ciphertext: "FBSGECTOEKRZ"

Example2: Find cipher text in Playfair method.

Plaintext: "**The data is encrypted**"
Key: "**PRIVICY**"

Formatted Plaintext: "TH ED AT AI SE NC RY PT ED"

P	**R**	**I**	**V**	**C**
Y	**A**	**B**	**D**	**E**
F	**G**	**H**	**J**	**K**
L	**M**	**N**	**O**	**S**
T	**U**	**W**	**X**	**Z**

Ciphertext: "WFYEYUBRZKSIPAYPYE"

Example3: Find cipher text in Playfair method.

Plaintext: "**Authentication is important**"
Key: "**PROTECT**"

Formatted Plaintext: "AU TH EN TI CA TI ON IS IM PO RT AN TX"

P	**R**	**O**	**T**	**E**
C	**A**	**B**	**D**	**F**
G	**H**	**I**	**J**	**K**
L	**M**	**N**	**S**	**U**
V	**W**	**X**	**Y**	**Z**

Ciphertext: "FMRJOUOJABOJBXJNHNRTOEBMOY"

Bifid Cipher

The Bifid Cipher uses a Polybius Square to encipher a message in a way that makes it fairly difficult to decipher without knowing the secret. This is because each letter in the ciphertext message is dependent upon two letters from the plaintext message. As a result, frequency analysis of letters becomes much more difficult. The first step is to use the Polybius Square to convert the letters into numbers. We will be writing the numbers vertically below the message.

	1	2	3	4	5
1	p	h	q	g	m
2	e	a	y	l	n
3	o	f	d	x	k
4	r	c	v	s	z
5	w	b	u	t	i

Encryption using the above key:

Plaintext: "**defend the east wall of the castle**"

1- Plaintext is converted to its coordinates in the usual manner, but they are written vertically:

	d	**e**	**f**	**e**	**n**	**d**	**t**	**h**	**e**	**e**	**a**	**s**	**t**	**w**	**a**	**l**	**l**	**o**	**f**	**t**	**h**	**e**	**c**	**a**	**s**	**t**	**l**	**e**
Row:	**3**	**2**	**3**	**2**	**2**	**3**	**5**	**1**	**2**	**2**	**2**	**4**	**5**	**5**	**2**	**2**	**2**	**3**	**3**	**5**	**1**	**2**	**4**	**2**	**4**	**5**	**2**	**2**
Column:	**3**	**1**	**2**	**1**	**5**	**3**	**4**	**2**	**1**	**1**	**2**	**4**	**4**	**1**	**2**	**4**	**4**	**1**	**2**	**4**	**2**	**1**	**2**	**2**	**4**	**4**	**4**	**1**

2- Plaintext is then read out in rows:

3232235122245522233512424522 31215342112441244124212244 41

3- Plaintext is then divided up into pairs again:

32 32 23 51 22 24 55 22 23 35 12 42 45 22 31 21 53 42 11 24 41 24 41 24 21 22 44 41

4- The pairs turned back into letters using the table (Ciphertext):

f f y w a l i a y k h c z a o e u c p l r l r l e a s r

Since the first letter in the plaintext is encoded into the first and middle letters of the ciphertext, the recipient of the message must have the entire message before they can decode it. This means that if part of the ciphertext is discovered by a third party, it is unlikely that they will be able to crack it.
To decipher a Bifid encrypted message, user first convert each letter into its corresponding number via the Polybius Square. Now, divide the long string of numbers into two equal rows. The digit in the top row and the digit in the bottom row will together reference the decoded letter in the Polybius Square.

Example1: Find cipher text in Bifid method.

Plaintext: "**Data security**"

Keyword:

	1	2	3	4	5
1	b	g	w	k	z
2	q	p	n	d	s
3	i	o	a	x	e
4	f	c	l	u	m
5	t	h	y	v	r

1-

	D	**a**	**t**	**a**	**s**	**e**	**c**	**u**	**r**	**i**	**t**	**y**
Row:	**2**	**3**	**5**	**3**	**2**	**3**	**4**	**4**	**5**	**3**	**5**	**5**
Column:	**4**	**3**	**1**	**3**	**5**	**5**	**2**	**4**	**5**	**1**	**1**	**3**

2- "**235323445355 431355245113"**

3- "**23 53 23 44 53 55 43 13 55 24 51 13"**

4- **Ciphertext**: "**n y n u y r l w r d t w"**

Example2: Find cipher text in Bifid method.

Plaintext: "**The data is encrypted**"

Keyword:

	1	2	3	4	5
1	b	g	w	k	z
2	q	p	n	d	s
3	i	o	a	x	e
4	f	c	l	u	m
5	t	h	y	v	r

1-

	T	**h**	**e**	**d**	**a**	**t**	**a**	**i**	**s**	**e**	**n**	**c**	**r**	**y**	**p**	**t**	**e**	**d**
Row:	**5**	**5**	**3**	**2**	**3**	**5**	**3**	**3**	**2**	**3**	**2**	**4**	**5**	**5**	**2**	**5**	**3**	**2**
Column:	**1**	**2**	**5**	**4**	**3**	**1**	**3**	**1**	**5**	**5**	**3**	**2**	**5**	**3**	**2**	**1**	**5**	**4**

2- **"553235332324552532 1254313155325321 54"**

3- **"55 32 35 33 23 24 55 25 32 12 54 31 31 55 32 53 21 54"**

4- Ciphertext: "r o e a n d r s o g v i i r o y q v**"**

Example3: Find cipher text in Bifid method.

Plaintext: "**Authentication is important**"

Keyword:

	1	2	3	4	5
1	b	g	w	k	z
2	q	p	n	d	s
3	i	o	a	x	e
4	f	c	l	u	m
5	t	h	y	v	r

1-

	A	u	t	h	e	n	t	i	c	a	t	i	o	n	i	s	i	m	p	o	r	t	a	n	t
Row:	3	4	5	5	3	2	5	3	4	3	5	3	3	2	3	2	3	4	2	3	5	5	3	2	5
Column:	3	4	1	2	5	3	1	1	2	3	1	1	2	3	1	5	1	5	2	2	5	1	3	3	1

2- **"345532534353323234235 5325 34125311231123151522513 31"**

3- **"34 55 32 53 43 53 32 32 34 23 55 32 53 41 25 31 12 31 12 31 51 52 25 13 31"**

4- **Ciphertext**: "x r o y l y o o x n r o y f s i g i g i t h s w i"

Trifid Cipher

The Trifid Cipher is the Bifid Cipher taken to one more dimension. Instead of using a 5x5 Polybius Square, you use a 3x3x3 cube. Otherwise everything else remains the same. As with the Bifid Cipher, the cube can be mixed to add an extra layer of protection, but for these examples we not be using a mixed alphabet cube.

Keyword:"**EPSDUCVWYM.ZLKXNBTFGORIJHAQ"**

Layer1				**Layer2**				**Layer3**			
	1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
1	**E**	**P**	**S**	**1**	**M**	**.**	**Z**	**1**	**F**	**G**	**O**
2	**D**	**U**	**C**	**2**	**L**	**K**	**X**	**2**	**R**	**I**	**J**
3	**V**	**W**	**Y**	**3**	**N**	**B**	**T**	**3**	**H**	**A**	**Q**

The first step is to use the cube to convert the letters into numbers. We will be writing the numbers vertically below the message in the order of **Layer**, **Row, Column**.

Plaintext: "**DEFEND THE EAST WALL OF THE CASTLE."**

The first step means locating the plaintext letters in the layers above, "D" is in layer 1, row 2, column 1, so "D" becomes 121. In the same manner, "E" becomes 111. If we write down the numbers corresponding to each letter vertically, it becomes:

1- Plaintext is converted to its coordinates in the usual manner, but they are written vertically:

	D	**E**	**F**	**E**	**N**	**D**	**T**	**H**	**E**	**E**	**A**	**S**	**T**	**W**	**A**	**L**	**L**	**O**	**F**	**T**	**H**	**E**	**C**	**A**	**S**	**T**	**L**	**E**	**.**
Layer:	**1**	**1**	**3**	**1**	**2**	**1**	**2**	**3**	**1**	**1**	**3**	**1**	**2**	**1**	**3**	**2**	**2**	**3**	**3**	**2**	**3**	**1**	**1**	**3**	**1**	**2**	**2**	**1**	**2**
Row:	**2**	**1**	**1**	**1**	**3**	**2**	**3**	**3**	**1**	**1**	**3**	**1**	**3**	**3**	**3**	**2**	**2**	**1**	**1**	**3**	**3**	**1**	**2**	**3**	**1**	**3**	**2**	**1**	**1**
Column:	**1**	**1**	**1**	**1**	**1**	**1**	**3**	**1**	**1**	**1**	**2**	**3**	**3**	**2**	**2**	**1**	**1**	**3**	**1**	**3**	**1**	**1**	**3**	**2**	**3**	**3**	**1**	**1**	**2**

2- Plaintext is then read out in rows:

"113121231131213223323113122121
211132331131333221133123132111
111113111233221131311323311 2"

At the moment this is still a substitution cipher and fairly easy to break. The next step is to use a 'period', which is a number usually 5-20, which is part of the key material agreed on by both sender and receiver. If we take a period of 5,

DEFEN	**DTHEE**	**ASTWA**	**LLOFT**	**HECAS**	**TLE.**
11312	**12311**	**31213**	**22332**	**31131**	**2212**
21113	**23311**	**31333**	**22113**	**31231**	**3211**
11111	**13111**	**23322**	**11313**	**11323**	**3112**

3- We group the numbers. We now read off the numbers in each group horizontally, and do the substitution back to letters using the original keysquare.

113 122 111 311 111	**123 112 331 113 111**	**312 133 133 323 322**
S U E F E	**C P H S E**	**G Y Y J I**

223 322 211 311 313 311 313 123 111 323 221 232 113 112
X I M F O F O C E J L B S P

Ciphertext: "**SUEFE CPHSE GYYJI XIMFO FOCEJ LBSP**"

Which means "DEFEND THE EAST WALL OF THE CASTLE." is enciphered to "SUEFE CPHSE GYYJI XIMFO FOCEJ LBSP" using the key square above and a period of 5. To decipher a Trifid encrypted message, you first convert each letter into its corresponding number via the cube. Now, divide the long string of numbers into three equal rows. Now, read off each column and use the cube to convert the three numbers into the plaintext letter.

Example1: Find cipher text in Trifid method using period of 5.
Plaintext: "**Data security**"

Keyword: "**EPSDUCVWYM.ZLKXNBTFGORIJHAQ"**

Layer1				**Layer2**				**Layer3**			
	1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
1	**E**	**P**	**S**	**1**	**M**	**.**	**Z**	**1**	**F**	**G**	**O**
2	**D**	**U**	**C**	**2**	**L**	**K**	**X**	**2**	**R**	**I**	**J**
3	**V**	**W**	**Y**	**3**	**N**	**B**	**T**	**3**	**H**	**A**	**Q**

1-

	D	**a**	**t**	**a**	**s**	**e**	**c**	**u**	**r**	**i**	**t**	**y**
Layer:	**1**	**3**	**2**	**3**	**1**	**1**	**1**	**1**	**3**	**3**	**2**	**1**
Row:	**2**	**3**	**3**	**3**	**1**	**1**	**2**	**2**	**2**	**2**	**3**	**3**
Column:	**1**	**2**	**3**	**2**	**3**	**1**	**3**	**2**	**1**	**2**	**3**	**3**

2- "132311113321
233311222233
123231321233"

Datas ecuri ty

13231 11133 21
23331 12222 33
12323 13212 33

3-

132 312 333 112 323 111 331 222 213 212 213 333
W G Q P J E H K Z . Z Q

Ciphertext: "WGQPJ EHKZ.ZQ"

Example2: Find cipher text in Trifid method using period of 5.
Plaintext: "**The data is encrypted**"

Keyword: "**EPSDUCVWYM.ZLKXNBTFGORIJHAQ"**

Layer1				**Layer2**				**Layer3**			
	1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
1	**E**	**P**	**S**	**1**	**M**	**.**	**Z**	**1**	**F**	**G**	**O**
2	**D**	**U**	**C**	**2**	**L**	**K**	**X**	**2**	**R**	**I**	**J**
3	**V**	**W**	**Y**	**3**	**N**	**B**	**T**	**3**	**H**	**A**	**Q**

1-

	T h e d a t a i s e n c r y p t e d
Layer:	**2 3 1 1 3 2 3 3 1 1 2 1 3 1 1 2 1 1**
Row:	**3 3 1 2 3 3 3 2 1 1 3 2 2 3 1 3 1 2**
Column:	**3 1 1 1 2 3 2 2 3 1 1 3 1 3 2 3 1 1**

2- "231132331121311211
331233321132231312
311123223113132311"

Theda taise ncryp ted
23113 23311 21311 211
33123 33211 32231 312
31112 32231 13132 311

3- 231 133 312 331 112 233 113 321 132 231 213 113 223 113 132 211 312 311
N Y G H P T S R W N Z S X S W M G F

Ciphertext: "**NYGHP TSRWN ZSXSW MGF**"

Example3: Find cipher text in Trifid method using period of 5.
Plaintext: "**Authentication is important**"

Keyword: "**EPSDUCVWYM.ZLKXNBTFGORIJHAQ"**

Layer1				**Layer2**				**Layer3**			
	1	**2**	**3**		**1**	**2**	**3**		**1**	**2**	**3**
1	**E**	**P**	**S**	**1**	**M**	**.**	**Z**	**1**	**F**	**G**	**O**
2	**D**	**U**	**C**	**2**	**L**	**K**	**X**	**2**	**R**	**I**	**J**
3	**V**	**W**	**Y**	**3**	**N**	**B**	**T**	**3**	**H**	**A**	**Q**

1-

	A u t h e n t i c a t i o n i s i m p o r t a n t
Layer:	**3 1 2 3 1 2 2 3 1 3 2 3 3 2 3 1 3 2 1 3 3 2 3 2 3**
Row:	**3 2 3 3 1 3 3 2 2 3 3 2 1 3 2 1 2 1 1 1 2 3 3 3 3**
Column:	**2 2 3 1 1 1 3 2 3 2 3 2 3 1 2 3 2 1 2 3 1 3 2 1 3**

2- "312312231323323132133232332332133223321321211123333
22311132323231232123132 13"

Authe	**ntica**	**tioni**	**simpo**	**rtant**
31231	**22313**	**23323**	**13213**	**32323**
32331	**33223**	**32132**	**12111**	**23333**
22311	**13232**	**32312**	**32123**	**13213**

3- 312 313 233 122 311 223 133 322 313 232 233 233 213 232 312
G O T U F X Y I O B T T Z B G

132 131 211 132 123 323 232 333 313 213
W V M W C J B Q O Z

Ciphertext: "GOTUF XYIOB TTZBG WVMWC JBQOZ"

H.W:

1- In Playfair Ciphers, find encryptions from plaintexts:
"Bob sends code to Alice" using keyword "encrypt".
"Data security is important" using keyword "robust".
"Privacy is part form security" using keyword "integrity".

2- In Playfair Cipher, find decryptions from ciphertexts:
"SJBMBLTBMLRNHACUPCBJECRUECOMHBKZ" using keyword "secrecy".
"TDJRWCYEYGBILETAUDTIEDMDGDTITY" using keyword "hide".
"IEOJUDTOCIBHRSCFMDLY" using keyword "code".

3- In Bifid Cipher, find ciphertexts from plaintexts:
"Confidentiality means secrecy requirement"
"Electronic applications need data security"
"Each network has different services"

	1	2	3	4	5
1	g	h	q	p	c
2	e	n	y	l	a
3	k	f	d	x	o
4	w	m	i	s	z
5	r	b	u	t	v

4- In Bifid Cipher, find decryptions from ciphertexts:
"rmstwluhpzcgfqyvkaqk"
"ooewneqtdsetzwaghbgwvdpwaop"
"sdhaisiaqmgcibxoy"

	1	2	3	4	5
1	r	h	q	p	a

2	k	m	z	l	c
3	e	f	d	x	o
4	w	n	i	s	y
5	g	b	u	t	v

5- In Trifid Cipher, find encryptions from plaintexts:
"Confidentiality means secrecy requirement" using period of 5
"Electronic applications need data security" using period of 5
"One of the security methods is integration" using period of 5

Layer1				Layer2				Layer3			
	1	2	3		1	2	3		1	2	3
1	O	S	P	1	N	V	Z	1	F	G	E
2	J	C	L	2	U	D	R	2	X	I	K
3	Q	Y	W	3	M	B	T	3	A	H	.

6- In Trifid Cipher, find decryptions from ciphertexts:
"gervghgcwithncpbbpwqvhwasekuv." using period of 5
"ejv.kvgsmcwmnind.umhjxnub" using period of 5
"qucrzxg.kzktnpepqjiumbytsbiqwqt" using period of 5

Layer1				Layer2				Layer3			
	1	2	3		1	2	3		1	2	3
1	O	S	P	1	N	V	Z	1	F	G	E
2	J	C	L	2	U	D	R	2	X	I	K
3	Q	Y	W	3	M	B	T	3	A	H	.

Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 7

Contents

Transposition Ciphers

This cipher is a simple data encryption method (classical cipher) in which message characters are rearranged in some regular plaintext pattern to ciphertext. It is designed to confuse the attacker, however; it can be deciphered by the intended recipient. Unlike substitution ciphers that replace letters with other letters, a transposition cipher keeps the letters the same, but rearranges their order according to a specific algorithm. There are many algorithms for Transposition Ciphers such as Keyless, Rail Fence, Route Cipher, Columnar Transposition, Double Column Transposition, Keyed Transposition Ciphers.

Keyless Transposition Ciphers

Simple transposition ciphers, which were used in the past, are keyless. A good example of a keyless cipher using the first method is the rail fence cipher. The ciphertext is created reading the pattern row by row.

Example: Bob sends the message to Alice.
Plaintext: "**The data is encrypted**"

T		E		A		A		S		N		R		P		E	
	H		D		T		I		E		C		Y		T		D

Ciphertext: "**TEAASNRPEHDTIECYTD**".

To decipher, write half the letters on one line, half on the second.

Rail Fence

In the rail fence cipher, the plaintext is written downwards on successive "rails" of an imaginary fence, starting a new column when the bottom is reached. The message is then read off in rows.
Example: if we have 3 rails (notice we can use 4, 5, …etc.)
Plaintext: "**The data is encrypted**"
we would rearrange out:

T				A				S				R				E		
	H		D		T		I		E		C		Y		T		D	
		E				A				N				P				X

Note that at the end of the message we have inserted "X". The last "X" is just a random letter to fill in the space.

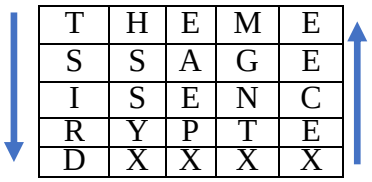
Ciphertext: "TASREHDTIECYTDEANP"

To decipher a message, we must know the number of rails that were used to encipher it. Then, we break up the letters into equal groups for each rail. For instance, if we are using 3 rails, we would break the secret message into 3 groups (depending on the zigzag line). Next, we use the table to rearrange the letters and get the plaintext.

Route Cipher

In this cipher, the plaintext is written in a grid, and then read off following the route chosen. We write the message vertically in columns, but instead of reading off the secret message horizontally, we read it off using a predetermined pattern.

Example: Bob sends the message to Alice. We have to decide on the number of columns, we will use 5 columns.
Plaintext: "**The message is encrypted**"



T	H	E	M	E
S	S	A	G	E
I	S	E	N	C
R	Y	P	T	E
D	X	X	X	X

Ciphertext: **"TSIRD HSSYX EAEPX MGNTX EECEX"**

The four "Xs" are just a random letter to fill in the space. With using a route of spiraling inwards counterclockwise from the bottom right we get:

"XECEE MEHT SIRD XXXT NGAS SYPE"

The complex pattern makes this algorithm difficult to decipher unless you know the key.

Also, there is a third type which is the clockwise cipher process.

H.W:

1- In Keyless Transposition Ciphers, find ciphertexts from plaintexts:

"Bob sends code to Alice"

"Data security is important"

Then find the deciphertext from obtained ciphertext

2- In Rail Fence, find ciphertexts from plaintexts:

"Client message transferred to the server" (use 4 key/rails)

"This is authentication code" (use 3 key/rails)

Then find the deciphertext from obtained ciphertext

3- In Route Cipher, find ciphertexts from plaintexts:

"Confidentiality means secrecy requirement" (use 4 columns)

"Electronic applications need data security" (use 5 columns)

Then find the deciphertext from obtained ciphertext

Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 8

Contents

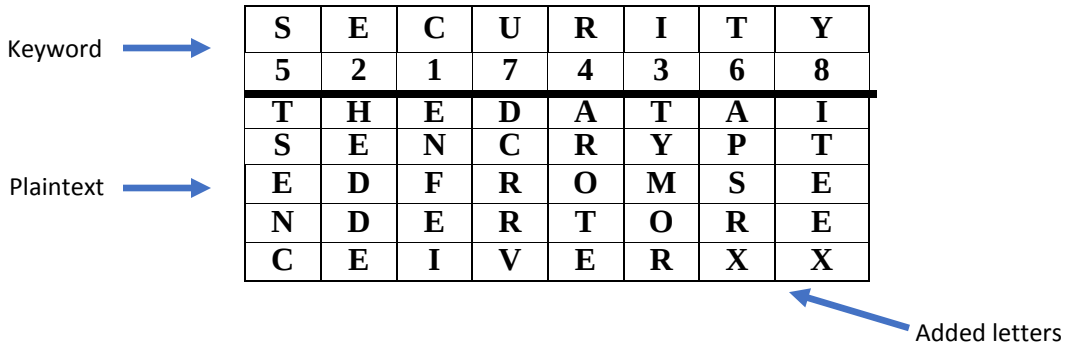
Columnar Transposition

In a columnar transposition, the message is written out in rows of a fixed length. The message is then read out by column by column, where the columns are chosen in some scrambled order. The number of columns and the order in which they are chosen is defined by a keyword. For example, the word "SECURITY" is 8 letters long. Therefore, there are 8 columns that will be read of in the following order: 5 2 1 7 4 3 6 8. The order is chosen by the alphabetical order of the letters in the keyword. If a letter is repeated, we do the one that appears first, then the next and so on. There are two cases which are regular and irregular.

Regular Case

In a regular columnar transposition cipher, the empty spaces are filled with random letters.

Example: Suppose we use the keyword "**SECURITY**"
Plaintext: "**The data is encrypted from sender to receiver**".
we would rearrange out the grid:



S	E	C	U	R	I	T	Y
5	2	1	7	4	3	6	8
T	H	E	D	A	T	A	I
S	E	N	C	R	Y	P	T
E	D	F	R	O	M	S	E
N	D	E	R	T	O	R	E
C	E	I	V	E	R	X	X

Ciphertext: "**ENFEI HEDDE TYMOR AROTE TSENC APSRX DCRRV ITEEX**"
The eight columns wrote out in the scrambled order defined by the keyword "SECURITY".

Decipher text:
1- We start to write out the keyword on the grid then and arrange the alphabetical order of the letters of the keyword.

S	E	C	U	R	I	T	Y
5	2	1	7	4	3	6	8

2- We must divide the length of the ciphertext by the length of the keyword. After that, we write the ciphertext down the first column until you reach the last row.

Ciphertext: "**ENFEIHEDDETYMORAROTETSENCAPSRXDCRRVITEEX**"

: "**ENFEI HEDDE TYMOR AROTE TSENC APSRX DCRRV ITEEX**"

S	E	C	U	R	I	T	Y
5	2	1	7	4	3	6	8
		E					
		N					
		F					
		E					
		I					

Irregular Case

In the irregular case, the empty spaces are not filled in with random letters:

Example: Suppose we use the keyword "**SECURITY**"

Plaintext: "**The data is encrypted from sender to receiver**".
we would rearrange out the grid:

Keyword →

Plaintext →

S	E	C	U	R	I	T	Y
5	2	1	7	4	3	6	8
T	H	E	D	A	T	A	I
S	E	N	C	R	Y	P	T
E	D	F	R	O	M	S	E
N	D	E	R	T	O	R	E
C	E	I	V	E	R		

Ciphertext: "**ENFEI HEDDE TYMOR AROTE TSENC APSR DCRRV ITEE**"

Decipher text: The use of the same previous two steps in the regular case as well as the following equations to find out the added letters.

Number of Ciphertext characters/number of keyword characters = rows number
Rows number * number of keyword characters = total value
The total value - number of ciphertext characters= the added characters.

$38/8=4.75 \approx 5$

$5 * 8= 40$

40- 38= 2 (This means that there are two nulls in the grid.)

Double Column Transposition

This algorithm applies a columnar transposition twice to prevent decipher by attackers. Double column takes the result of the irregular columnar transposition in the previous section and perform a second encryption with a different keyword such as "PRIVACY" which gives the order "4 5 3 6 1 2 7".

Example: Suppose we use the first keyword "SECURITY" and second keyword "**PRIVACY**"
Plaintext: "**The data is encrypted from sender to receiver**".

1- We would rearrange out plaintext on the grid with first keyword "SECURITY":

Keyword →
Plaintext →

S	E	C	U	R	I	T	Y
5	2	1	7	4	3	6	8
T	H	E	D	A	T	A	I
S	E	N	C	R	Y	P	T
E	D	F	R	O	M	S	E
N	D	E	R	T	O	R	E
C	E	I	V	E	R		

Ciphertext1: "**ENFEI HEDDE TYMOR AROTE TSENC APSR DCRRV ITEE**"

2- We would rearrange out ciphertext1 on the grid with first keyword "PRIVACY":

Keyword →
Plaintext →

P	R	I	V	A	C	Y
4	5	3	6	1	2	7
E	N	F	E	I	H	E
D	D	E	T	Y	M	O
R	A	R	O	T	E	T
S	E	N	C	A	P	S
R	D	C	R	R	V	I
T	E	E				

Ciphertext2: "**IYTAR HMEPV FERNCE EDRSRT NDAEDE ETOCR EOTSI**"

Keyed Transposition Ciphers

The keyless ciphers permute the characters by using writing plaintext in one way and reading it in another way. The permutation is done on the whole plaintext to create the whole ciphertext. Another method is to divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.

Example:

Plaintext: Alice sends the message "**The data is encrypted" to Bob**.

1- we divide the plaintext into blocks of predetermined size and independently permute each block.

2- Both sender and receiver previously agree to have the size of the block such as 5
"THEDA DAISE NCRYP TED"

3- The last two letters "XX" is to complete the block size of 5.
"THEDA DAISE NCRYP TEDXX"

4- Sender and Receiver use the following key for encryption and decryption (or different key):

Plaintext

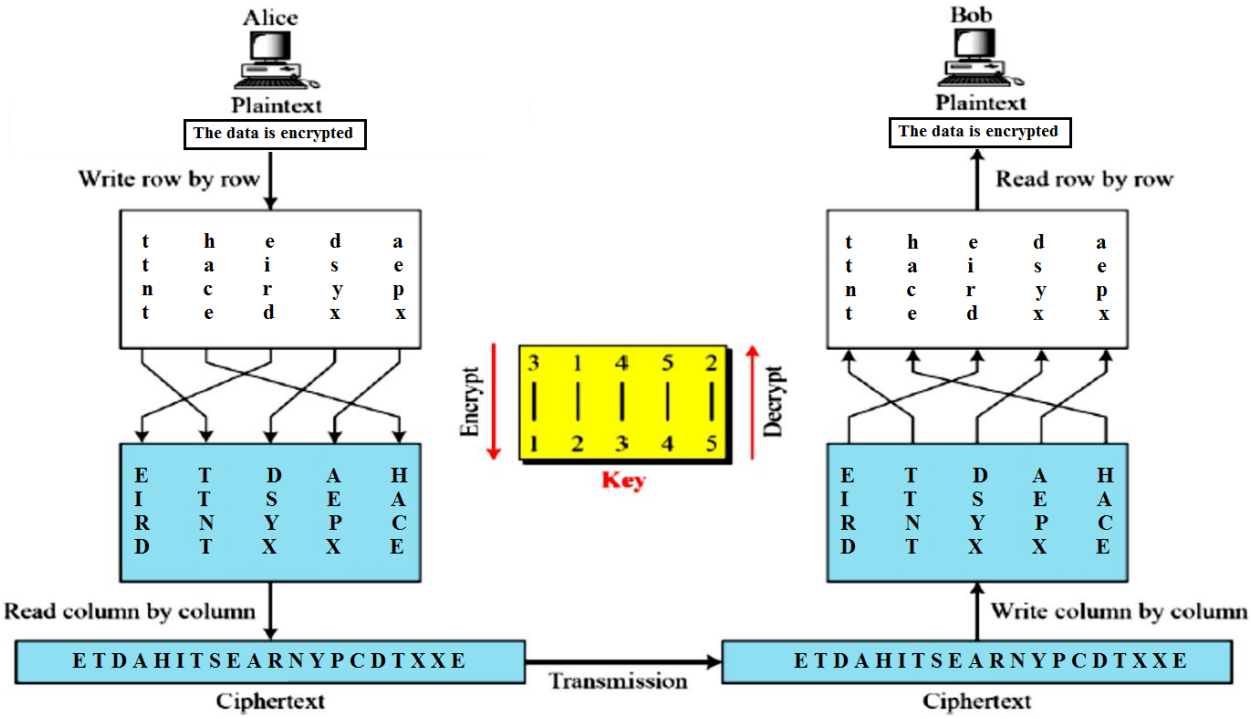
T	H	E	D	A
T	A	I	S	E
N	C	R	Y	P
T	E	D	X	X

Ciphertext

E	T	D	A	H
I	T	S	E	A
R	N	Y	P	C
D	T	X	X	E

Ciphertext: "ETDAH ITSEA RNYPC DTXXE"

The following figure illustrates the encryption and decryption operations for the previous example.



H.W:

1- In Columnar Transposition Ciphers, find ciphertexts from plaintexts in regular and irregular cases:

Plaintext: “The attacker tries to eavesdrop on user data” keyword: “protection”

Plaintext: “Authentication is the first phase in network security” keyword: “secrecy”

Then find the deciphertext from obtained ciphertext

2- In Double Column, find ciphertexts from plaintexts:

Plaintext: “User data transferred to the database” first keyword: “encrypt” second keyword: “encode”

Plaintext: “Weak security measures are compromised” 1st keyword: “authorize” 2sd keyword: “private”

Then find the deciphertext from obtained ciphertext

3- In Keyed Cipher, find ciphertexts from plaintexts:

Plaintext: “Integrity means preventing data modification” the block size of 5 and the key is below:

2	**4**	**1**	**3**	**5**
1	**2**	**3**	**4**	**5**

Plaintext: “Electronic applications need data security” the block size of 5 and the key is below:

5	**4**	**3**	**2**	**1**
1	**2**	**3**	**4**	**5**

Then find the deciphertext from obtained ciphertext

Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 9

Contents

Private Key

Block Cipher

Block Cipher - An encryption scheme that "the clear text is broken up into blocks of fixed length, and encrypted one block at a time" (such as DES, AES, IDEA, Blowfish algorithm). Usually, a block cipher encrypts a block of clear text into a block of cipher text of the same length. In this case, a block cipher can be viewed as a simple substitute cipher with character size equal to the block size. Main properties of this type:

- Identical clear text blocks are encrypted to identical cipher text blocks.
- Re-ordering clear text blocks results in re-ordering cipher text blocks.
- An encryption error affects only the block where it occurs.

Data Encryption Standard (DES)

The Data Encryption Standard (DES), known as the Data Encryption Algorithm (DEA), in the past, has been most widely used block cipher in world, especially in financial industry. It encrypts 64-bit data, and uses 56-bit key with 16 48-bit sub-keys.

DES (Data Encryption Standard) History

In the early 1970s, there are no cryptagraphic equipment available on the market. Although several small companies made and sold cryptographic equipment. The equipment was all different and could not interoperate. No one really knew if any of it was secure; there was no independent body to certify the security. In 1972, the National Bureau of Standards (NBS), now the National Institute of Standards and Technology (NIST), initiated a program to protect computer and communications data. They specified a series of design criteria:

- **Security.**
- **Completely specified and easy to understand.**
- **Public and available to all users.**

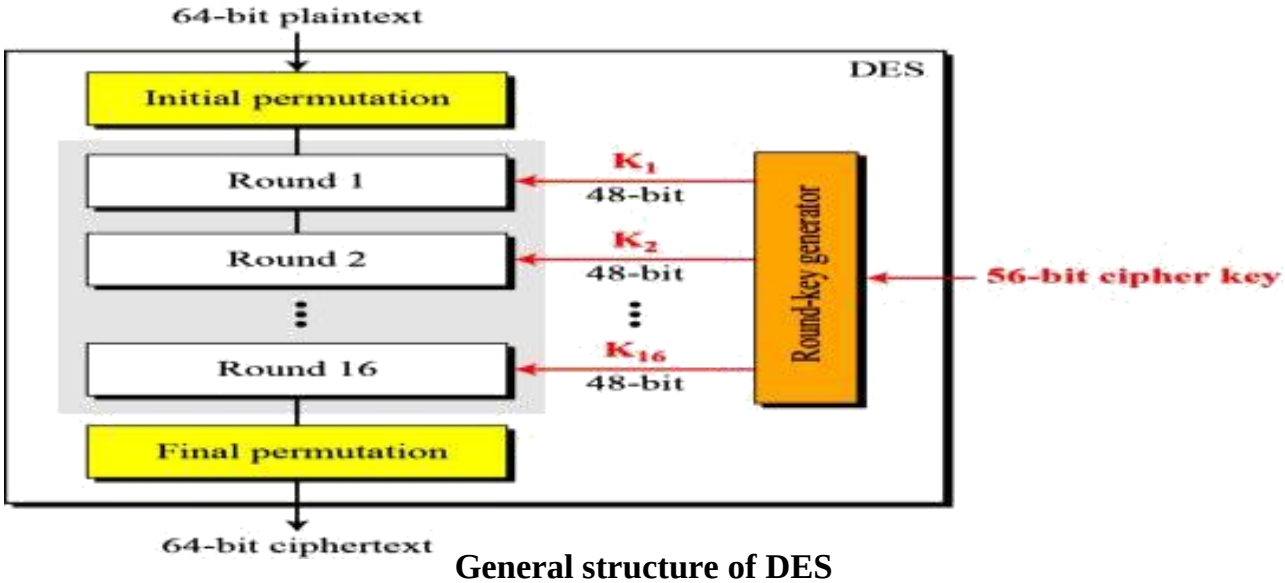
- **Efficient to use.**
- **Able to be validated.**
- **Exportable.**

Then, NIST proposed Triple DES that is a symmetric key-block cipher which applies the DES cipher in triplicate (3DES). DES's dominance came to an end in 2002, when the Advanced Encryption Standard (AES) replaced the DES encryption algorithm as the accepted standard.

Description of DES

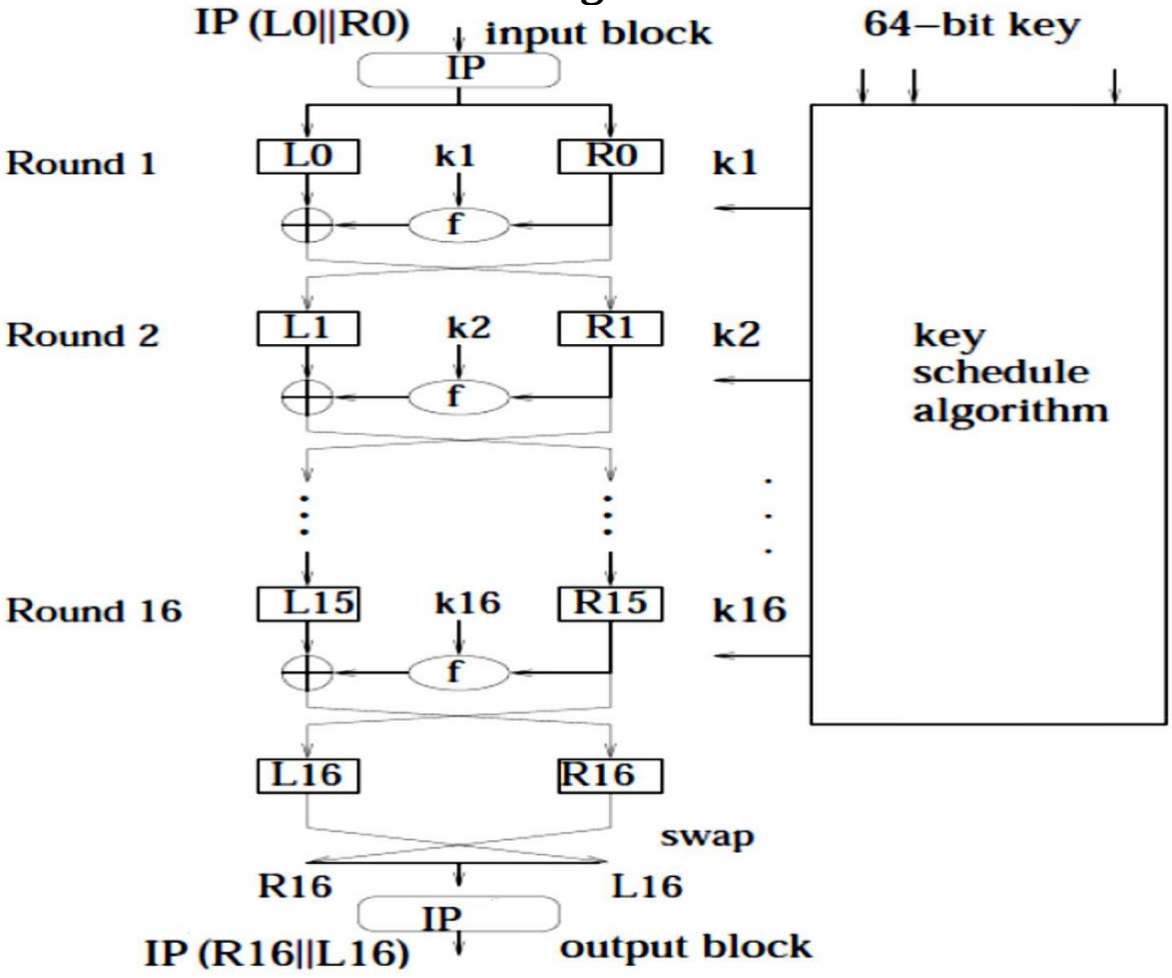
DES is a symmetric algorithm: The same algorithm and key are used for both encryption and decryption. DES is a block cipher; it encrypts data in 64-bit blocks. The key length is 56 bits. (The key is usually expressed as a 64-bit number, but every eighth bit is used for parity checking and is ignored. These parity bits are the least-significant bits of the key bytes.) The key can be any 56-bit number and can be changed at any time. All security rests within the key. At its simplest level, the algorithm is nothing more than a combination of the two basic techniques of encryption: confusion and diffusion. The fundamental building block of DES is a single combination of these techniques (a substitution followed by a permutation) on the text, based on the key. This is known as a round. DES has 16 rounds; it uses the same combination of techniques on the plaintext block 16 times.]

Encryption and decryption with DES



General structure of DES

DES Algorithm



Outline of the Algorithm

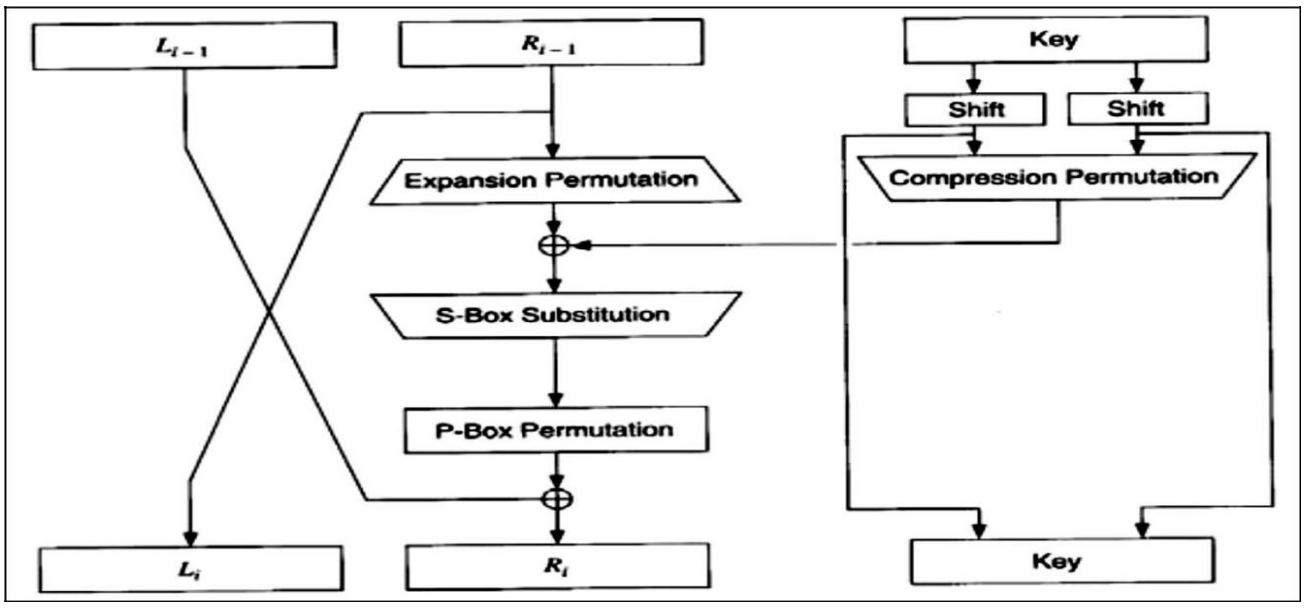
The basic process in enciphering a 64-bit data block using the DES consists of:

- An initial permutation (IP).
- 16 rounds of a complex key dependent calculation *f*.
- Final permutation, being the inverse of IP.

In each round the key bits are shifted, and then 48 bits are selected from the 56 bits of the key. The right half of the data is expanded to 48 bits via an expansion permutation, combined with 48 bits of a shifted and permuted key via an XOR, sent through 8 S-boxes producing 32 new bits, and permuted again. These four operations make up Function *f*. The output of Function *f* is then combined with the left half via another XOR. The result of these operations becomes the new right half; the old right half becomes the new left half.

If *Bi* is the result of the *ith* iteration, Li and Ri are the left and right halves of *Bi*, *Ki* is the 48-bit key for round *i*, and *f* is the function that does all the substituting and permuting and XORing with the key, then a round looks like:

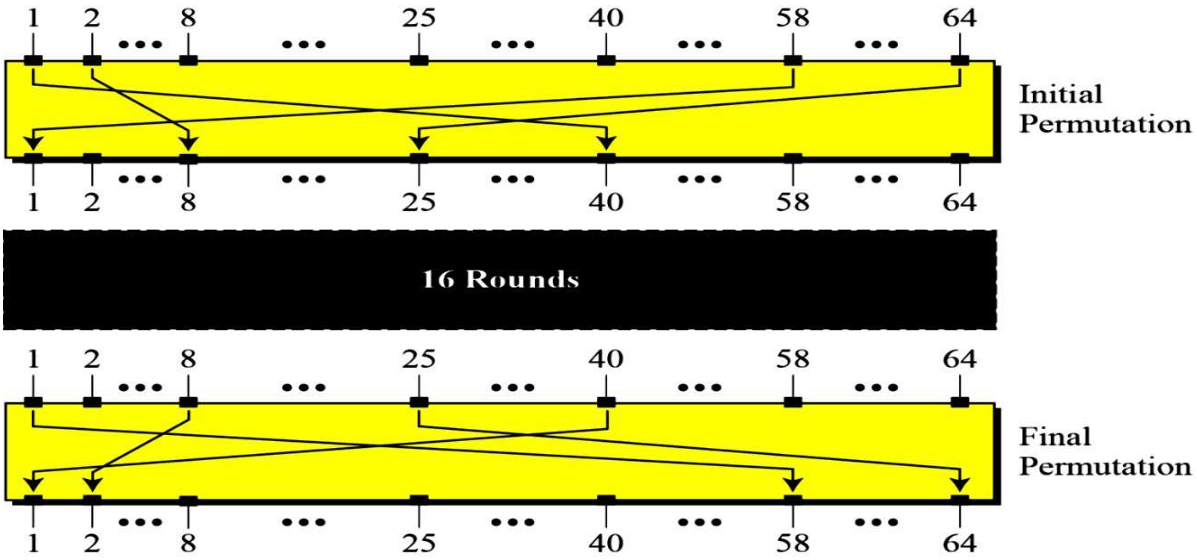
Li = R $_{i-1}$
Ri = L $_{i-1}$ **Xor f (R**$_{i-1}$**, K**$_i$**)**



A round in DES

Initial and final permutation steps in DES

Two permutations (P-boxes), initial and final permutations.



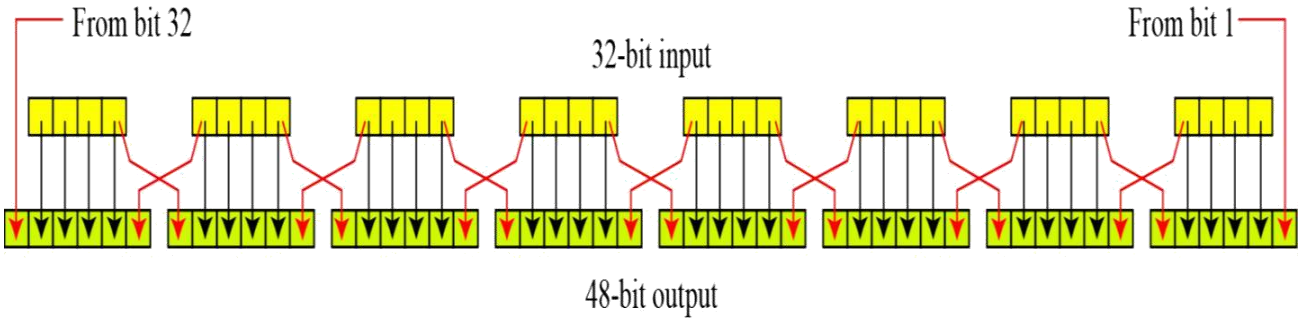
Initial and final permutation tables

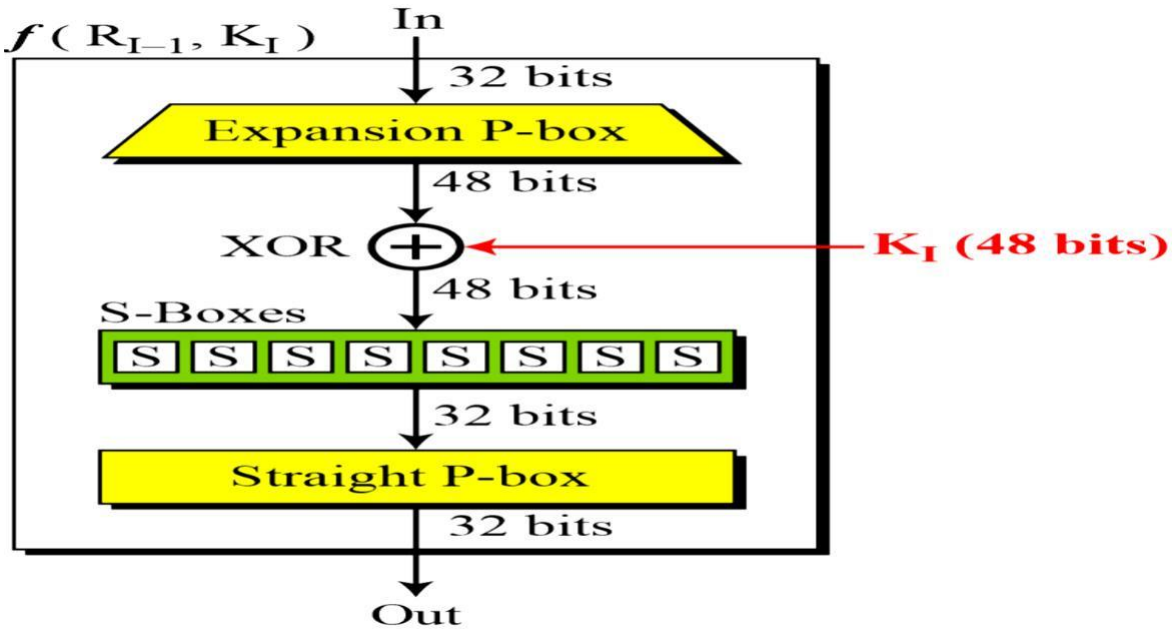
The initial permutation occurs before round 1; it transposes the input block as described in Table that should be read left to right, top to bottom. For instance, the initial permutation moves bit 58 of the plaintext to bit position 1, bit 50 to bit position 2, bit 42 to bit position 3, and so forth. The initial permutation and the corresponding final permutation do not improve DES's security, just make DES more complex.

Initial Permutation								*Final Permutation*							
58	50	42	34	26	18	10	02	40	08	48	16	56	24	64	32
60	52	44	36	28	20	12	04	39	07	47	15	55	23	63	31
62	54	46	38	30	22	14	06	38	06	46	14	54	22	62	30
64	56	48	40	32	24	16	08	37	05	45	13	53	21	61	29
57	49	41	33	25	17	09	01	36	04	44	12	52	20	60	28
59	51	43	35	27	19	11	03	35	03	43	11	51	19	59	27
61	53	45	37	29	21	13	05	34	02	42	10	50	18	58	26
63	55	47	39	31	23	15	07	33	01	41	09	49	17	57	25

DES Function

The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.





The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.

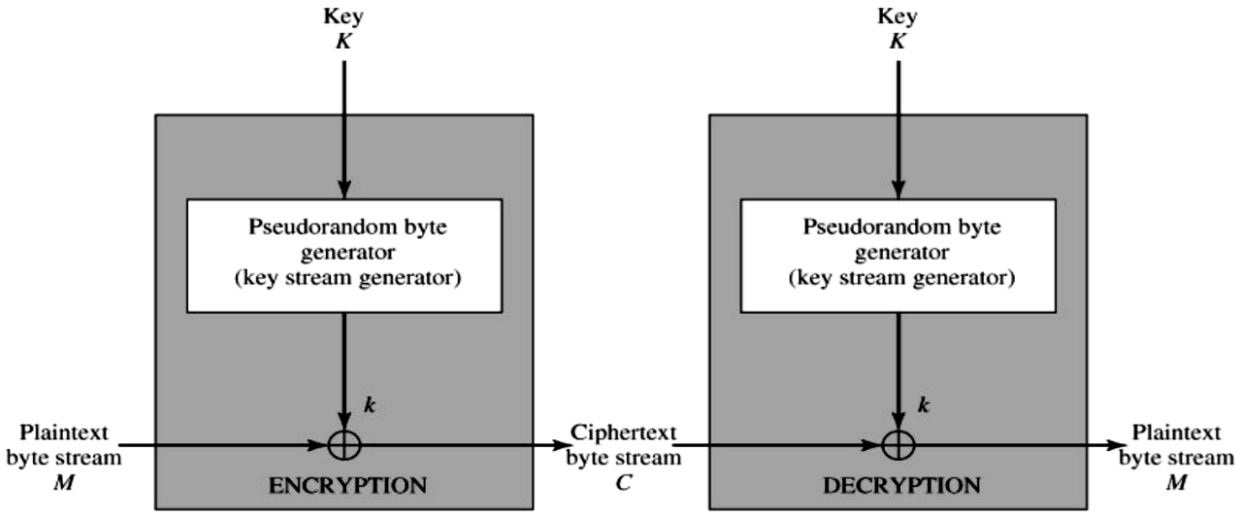
Stream Cipher

A typical stream cipher encrypts plaintext one byte at a time; although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time such as RC4, Grain, SEAL. In the following figure is a representative diagram of stream cipher structure. In this structure a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random. For now, we simply say that a pseudorandom stream is one that is unpredictable without knowledge of the input key. The output of the generator, called a keystream, is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation. For instance, if the next byte generated by the generator is 01101100 and the next plaintext byte is 11001100, then the resulting ciphertext byte is

```
  11001100   plaintext
⊕ 01101100   key stream
  10100000   ciphertext
```

Decryption requires the use of the same pseudorandom sequence:

```
  10100000   ciphertext
⊕ 01101100   key stream
  11001100   plaintext
```



The basic working principle of stream cipher.

Note: The stream cipher is similar to the one-time pad. The difference is that a one-time pad uses a genuine-random number stream, whereas a stream cipher uses a pseudo-random number stream.

Important elements for design a stream cipher

1. The encryption sequence should have a large period. A pseudorandom number generator uses a function that produces a deterministic stream of bits that eventually repeats. The longer the period of repeat the more difficult it will be to do cryptanalysis. This is essentially the same consideration that was discussed with reference to the Vigenère cipher, namely that the longer the keyword the more difficult the cryptanalysis.

2. The keystream should approximate the properties of a true random number stream as close as possible. For example, there should be an approximately equal numbers of 1s and 0s. If the keystream is treated as a stream of bytes, then all of the 256 possible byte values should appear approximately equally often. The more random-appearing the keystream is, the more randomized the ciphertext is, making cryptanalysis more difficult.

3. The output of the pseudorandom number generator is conditioned on the value of the input key. To guard against brute-force attacks, the key needs to be sufficiently long. The same considerations as apply for block ciphers are valid here. Thus, with current technology, a key length of at least 128 bits is desirable.

Pseudorandom Number Generator (PRNGs)

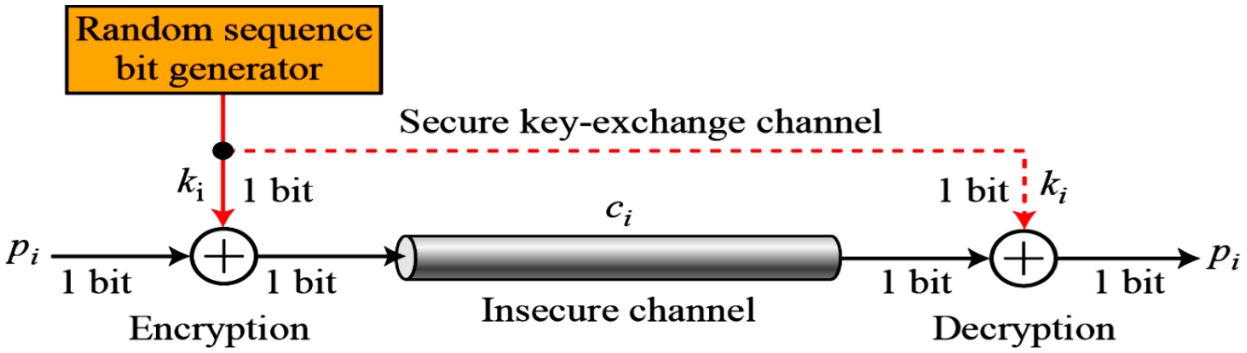
Cryptographic applications typically make use of algorithmic techniques for random number generation. These algorithms are deterministic and therefore produce sequences of numbers that are not statistically random. However, if the algorithm is good, the resulting sequences will pass many reasonable tests of randomness. Such numbers are referred to as pseudorandom numbers.

Types of stream ciphers

A stream cipher generates successive elements of the keystream based on an internal state. This state is updated in essentially two ways: if the state changes independently of the plaintext or ciphertext messages, the cipher is classified as a synchronous stream cipher. By contrast, self-synchronising stream ciphers update their state based on previous ciphertext digits.

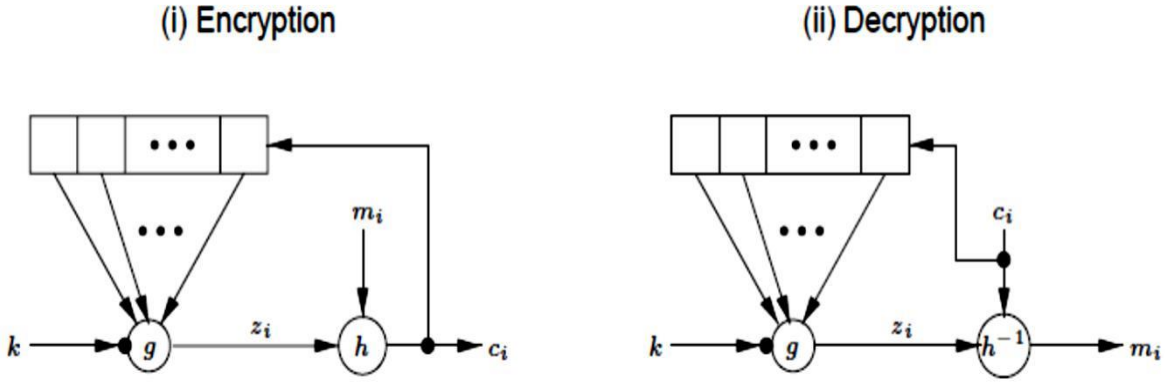
Synchronous stream ciphers

Definition: A synchronous stream cipher is one in which the keystream is generated independently of the plaintext message and of the ciphertext. The encryption process of a synchronous stream cipher.



Self-synchronizing stream ciphers

Definition: A self-synchronizing or asynchronous stream cipher is one in which the keystream is generated as a function of the key and a fixed number of previous ciphertext digits. The encryption function of a self-synchronizing stream cipher.



RC4 Algorithm

This stream cipher was invented in 1987 by Ron Rivest. Even though the RC4 cipher is officially named "Rivest Cipher 4", it is also known as "Ron's Code 4" (RC2, RC5 and RC6 also exist).

The trade secret behind RC4 was revealed in September 1994 when the description of the cipher was sent to the Cypherpunks mailing list (group of people interested in privacy and cryptography who used this mailing list to communicate). After that, the description was posted on many website and the genuineness of the information was confirmed as the resulting outputs of the described cipher were matching the outputs of licensed RC4. RC4 had a really large success thanks to its simplicity and efficiency. It was used in many popular standards and protocols such as WEP, WPA, SSL or TLS. Unfortunately, the cipher has some weaknesses and is not used anymore in modern protocols.

Principle

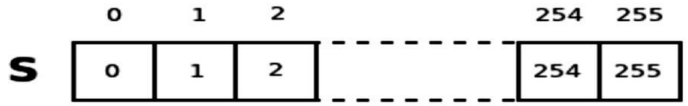
The RC4 algorithm generates a pseudo-random keystream that is then used to generate the ciphertext (by XORing it with the plaintext). It is called pseudo-random because it generates a sequence of numbers that only approximates the properties of random numbers. The sequence of bytes generated is not random since the output is always the same for a given input but it has to approximate random properties to make it harder to crack. The keystream is generated from a variable length key using an internal state composed of the following elements:

- A 256 bytes array (denoted S) containing a permutation of these 256 bytes.
- Two indexes i and j, used to point elements in the S array (only 8 bits are necessary for each index since the array only have 256 elements)

(A variable-length key of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S, with elements S[0], S[1], …, S[255]. At all times, S contains a permutation of all 8-bit numbers from 0 through 255). Once the S array has been initialized and "shuffled" with the key-scheduling algorithm (KSA), it is used and modified in the pseudo-random generation algorithm (PRGA) to generate the keystream.

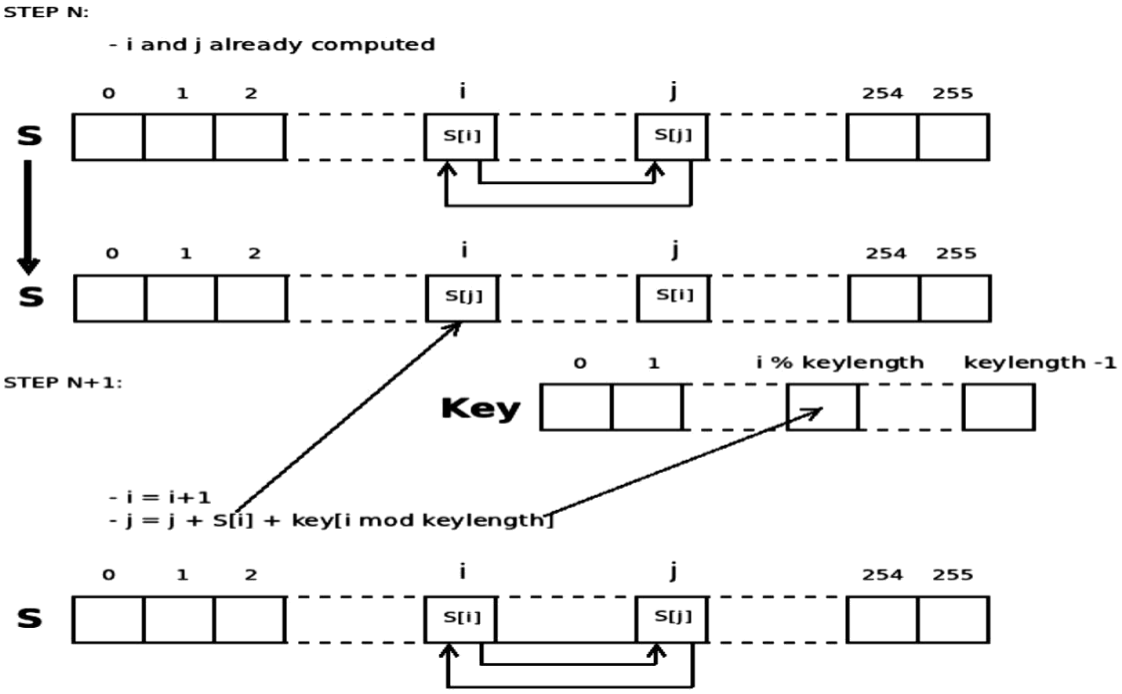
The key-scheduling algorithm

As explained before, the key-scheduling algorithm is used to generate the permutation array. The first step of this algorithm consist in initializing the S table with the identity permutation: the values in the array are equal to their index.



Once the S array is initialized, the next step consist in shuffling the array using the key to make it a permutation array. To do so, we simply iterate 256 times the following actions after initializing i and j to 0:

- **compute j = j + S[i] + key[i mod keylength]**
- **swap S[i] and S[j]**
- **increment i**



Once i has reached 256 (the 256 iterations were completed), the S array has been properly initialized. Here is some pseudo-code corresponding to the key-scheduling algorithm:

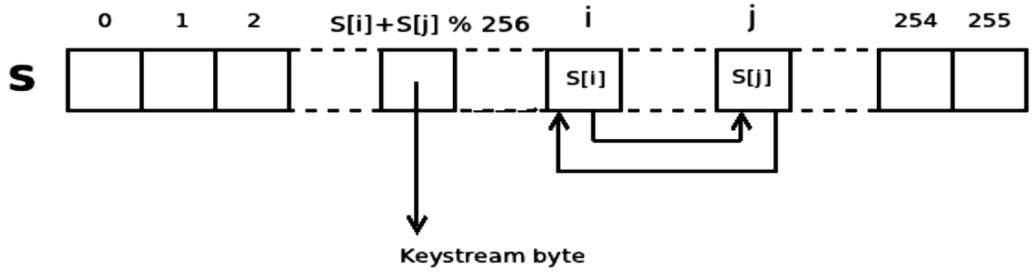
For i from 0 to 255
S[i] := i
End for
j := 0
For i from 0 to 255
j := (j + S[i] + key[i mod key length]) mod 256
Swap values of S[i] and S[j]
End for

Now that the S array is generated, it is used in the next step of the RC4 algorithm to generate the keystream.

The pseudo-random generation algorithm

This step of the algorithm consists in generating a keystream of the size of the message to encrypt. This algorithm enables us to generate a keystream of any size. To do so, we first initialize the two indexes to 0 and we then start the generation of the keystream one byte at a time until we reached the size of the message to encrypt. For each new byte to compute we do the following actions:

- **Compute new value of i and j:**
  i := (i + 1) % 256
  j := (j + S[i]) % 256
- **Swap S[i] and S[j] to have a dynamic state.**
- **Retrieve the next byte of the keystream from the S array at the index S[i]+S[j]% 256.**

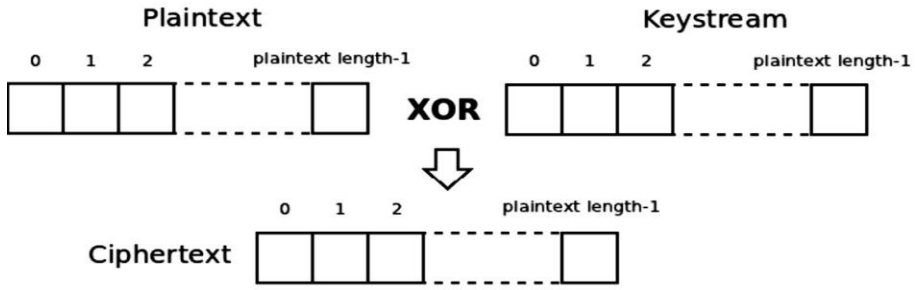


Here is some pseudo-code corresponding to the pseudo-random generation algorithm:

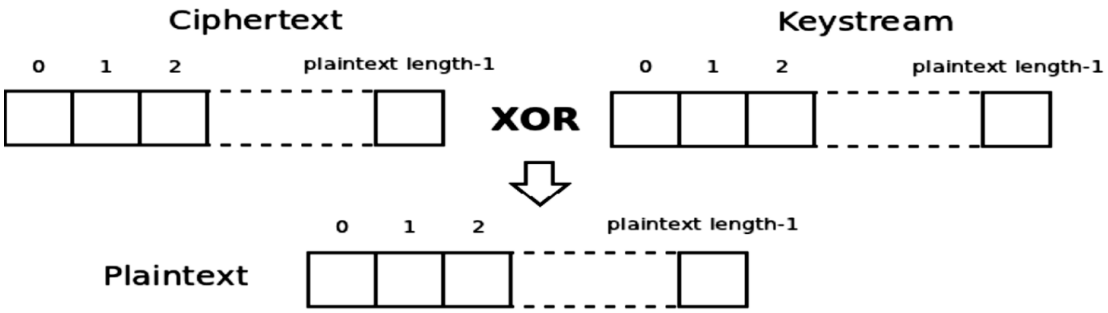
```
i := 0
j := 0
while Generating Output:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap values of S[i] and S[j]
    K := S[(S[i] + S[j]) mod 256]
    output K
end while
```

Encryption and decryption

Once the keystream has been generated, the encryption of the plaintext is really simple: it simply consists of a XOR between the plaintext and the keystream. See below an illustration of the encryption:



As for the decryption, it is as simple as the encryption, we only have to do the opposite: XOR the ciphertext with the keystream.



Data Security
4th Class

Lecturer: Dr. Mishall Hammed Awaad
Computer Science Department
Education College for Pure Sciences
University of Thi-Qar

Data Security
First Semester
Lecture 10

Contents

Public-Key Cryptosystems

Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristic:

- It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.

In addition, some algorithms, such as RSA, also exhibit the following characteristic:

- Either of the two related keys can be used for encryption, with the other used for decryption.

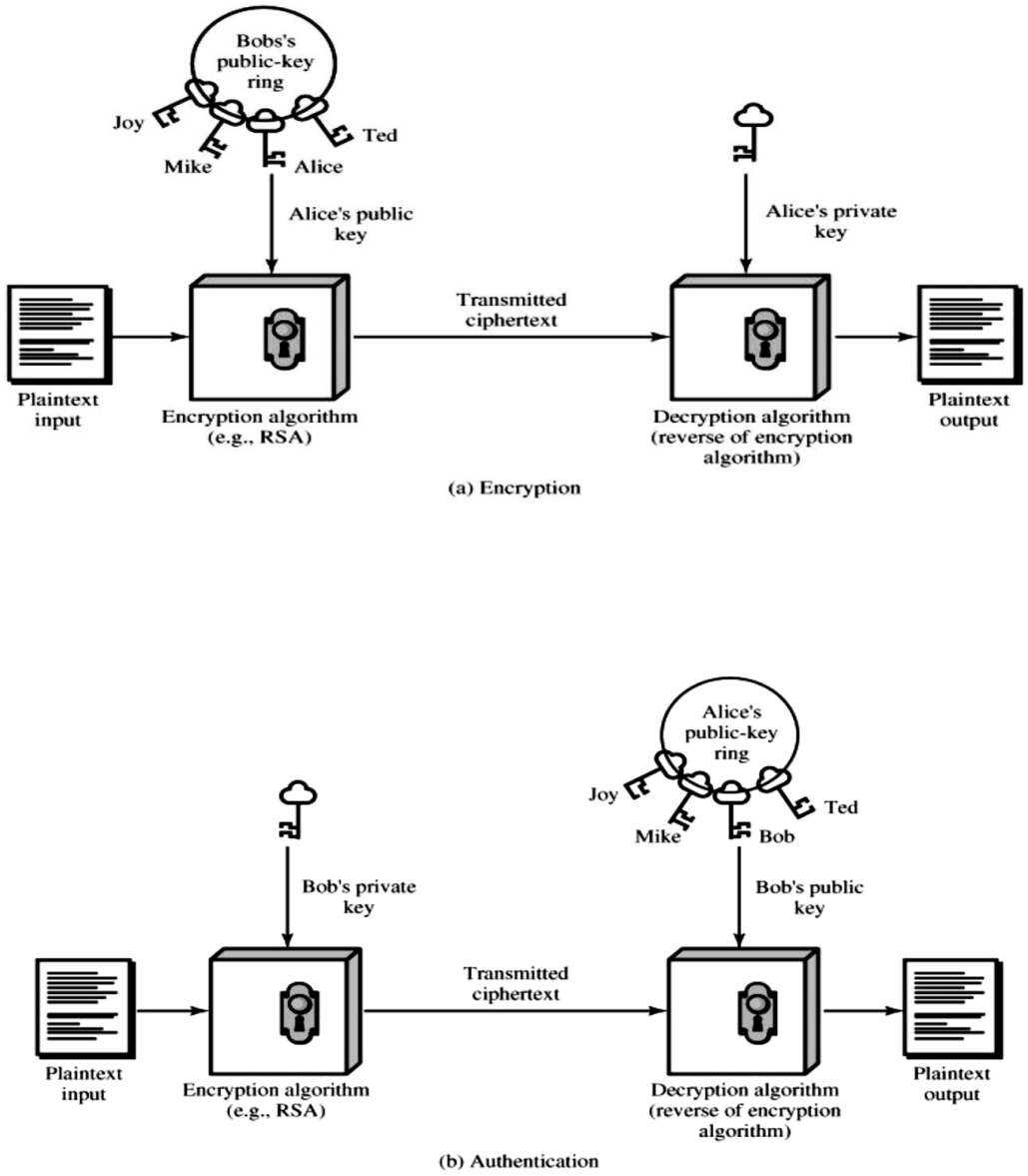


Figure: Public-Key Cryptography.

A public-key encryption scheme has six ingredients:

Plaintext: This is the readable message/data.
Encryption algorithm: It performs various transformations on the plaintext.

Public and private keys: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.

Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.
2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private, each user maintains a collection of public keys obtained from others.
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.
4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

With this approach, all participants have access to public keys. The private keys are generated locally by each participant and therefore need never be distributed. As long as a user's private key remains protected and secret, incoming communication is secure. At any time, a system can change its private key and publish the companion public key to replace its old public key.

In the following Table summarizes some of the important aspects of symmetric and asymmetric encryption. To discriminate between the two, we refer to the key used in symmetric encryption as a secret key. The two keys used for asymmetric encryption are referred to as the public key and the private key. Invariably, the private key is kept secret, but it is referred to as a private key rather than a secret key to avoid confusion with symmetric encryption.

Table: Conventional and Public-Key Encryption.

Symmetric Encryption	Asymmetric Encryption
Needed to Work:	Needed to Work:
1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key.	1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one).
Needed for Security:	Needed for Security:
1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Using below Figure (Public) compare with Figure (private). There is some source A that produces a message in plaintext, X =[X1, X2 ..., XM,]. The M elements of X are letters in some finite alphabet. The message is intended for destination B. B generates a related pair of keys: a public key, PU_b, and a private key, PU_b. PU_b is known only to B, whereas PU_b is publicly available and therefore accessible by A.

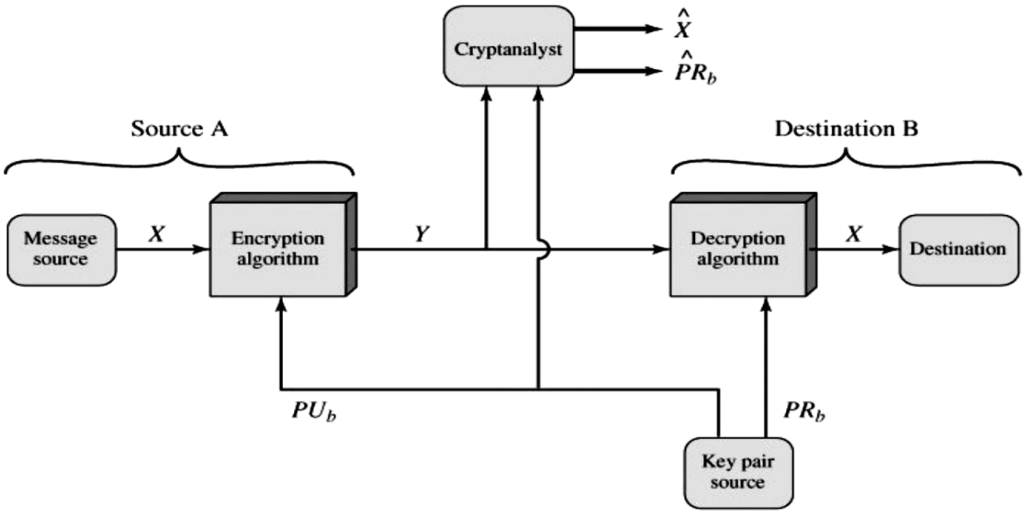


Figure: Public-Key Cryptosystem: Secrecy.

Applications for Public-Key Cryptosystems

Public-key systems are characterized by the use of a cryptographic algorithm with two keys, one held private and one available publicly. Depending on the application, the sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function. In broad terms, we can classify the use of public-key cryptosystems into three categories:

Encryption/decryption: The sender encrypts a message with the recipient's public key.

Digital signature: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

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

Table: Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

RSA Algorithm

The RSA cryptosystem, named after its inventors Ronald Rivest, Adi Shamir, and Leonard Adleman, is the most widely used public-key cryptosystem. It may be used to provide both secrecy and digital signatures and its security is based on the intractability of the fact that factorizing very large numbers is a `hard' problem. It was described to the public in August 1977. However, it was recently revealed that this method was originally invented in 1973.

RSA Initialization:

INPUT: A way to generate or select large random prime numbers.

OUTPUT: A public key, (***n, e***), and a private key, ***d***.

1. Select or generate two large random prime numbers, **p** and **q**.

2. Compute $n = p*q$ and $Ø = (p - 1)*(q - 1)$.

3. Select a random integer e, $1 < e < Ø$, such that $\gcd(e, Ø) = 1$.

4. Using the extended Euclidean Algorithm, find the unique integer **d;** $1 < d < Ø$, such that $e*d \equiv 1 \pmod{Ø}$.

5. Publish the public key, (***n, e***), and keep the private key, **d**, secret.

RSA Encryption:

INPUT: The plaintext to encrypt, and the receiving user's public key (**n, e**).

OUTPUT: The encrypted ciphertext.

User **Alice** sends the message to user **Bob**.

1. Using an agreed hash function, convert the plaintext into a unique integer **m** in the interval [**0, *n* - 1**]

2. compute $c = m^e \pmod{n}$ and send ***c*** to user **Bob.**

RSA Decryption:

INPUT: The received encrypted ciphertext and the receiver's private key ***d.***

OUTPUT: The original plaintext. User **Bob** receives the message from user **A**.

1. Use the private key to compute $m = c^d \pmod{n}$

2. Recover the plaintext by applying the inverse of the hash function from above Algorithm, returning the integer in the interval **[0; *n* - 1]** to the unique message it represents.

Note: must ***gcd(e,Øn)=1***.

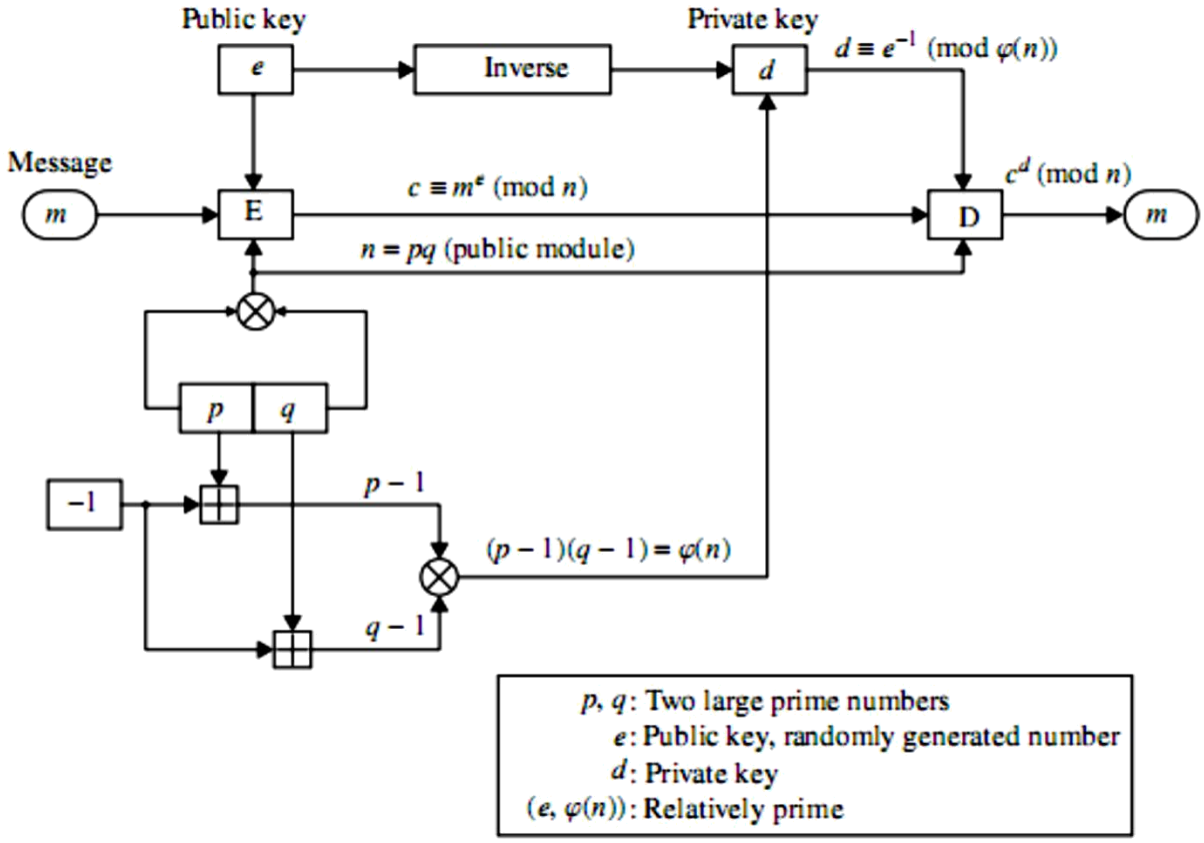


Figure: RSA public-key cryptosystem for encryption/decryption.