CRYPTOGRAPHY

Archana M

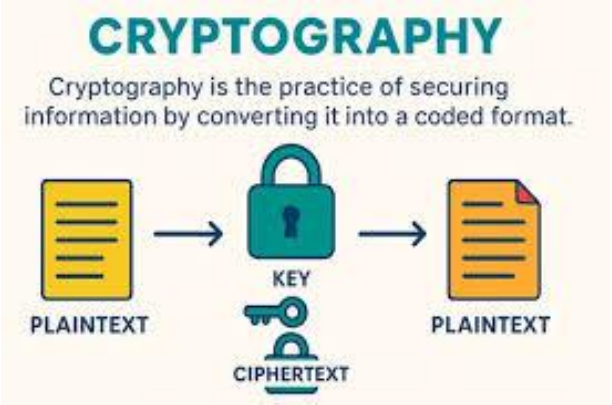
Department of Computer Applications

Introduction -Traditional Symmetric Key Ciphers

- **Cryptography** is the science and art of securing information by converting it into a form that is unreadable to unauthorized users.
- Its primary objective is to **protect data during storage and transmission** from unauthorized access, modification, or misuse.
- Cryptography assumes the presence of **adversaries** who may attempt to intercept or tamper with information.
- In the modern digital world, cryptography is the **foundation of secure communication systems**.

Introduction -Traditional Symmetric Key Ciphers

- It is widely used in **online banking, e-commerce, email systems, cloud computing, and blockchain technologies**.
- Whenever sensitive data such as **passwords, credit card details, or confidential messages** is transmitted over the internet, cryptographic techniques ensure security.
- These techniques guarantee that **only the intended recipient** can read and understand the transmitted information.



Introduction -Traditional Symmetric Key Ciphers

- SECURITY GOALS

Introduction -Traditional Symmetric Key Ciphers

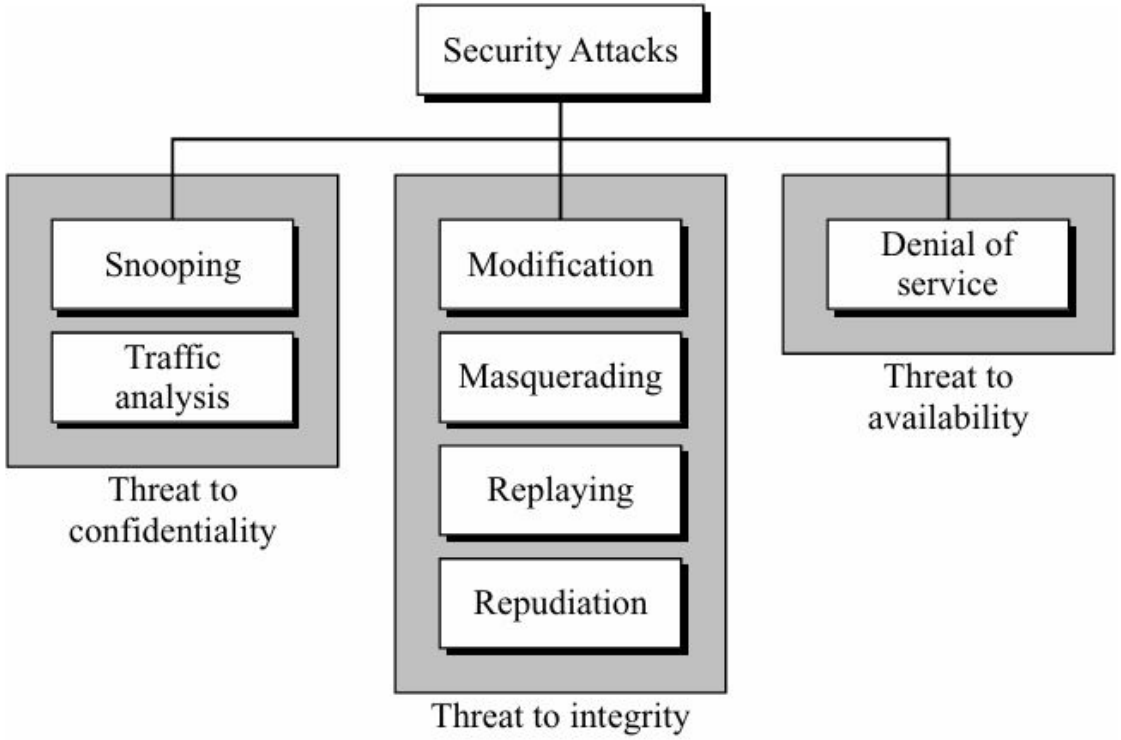
- **Confidentiality**
  - Ensures information is **kept secret** from unauthorized users
  - Protects sensitive data like **military secrets, business plans, and bank accounts**
  - Applies to both **stored data and data during transmission**
- **Integrity**
  - Ensures information is **accurate and not altered improperly**
  - Allows changes **only by authorized users and approved processes**
  - Prevents errors caused by **malicious attacks or system failures**

Introduction -Traditional Symmetric Key Ciphers

- **Availability**
  - Ensures information is **accessible when needed**
  - Allows authorized users to **use data without interruption**
  - Prevents harm caused by **system downtime or service denial**

Security Attacks

Security attacks threaten the three main goals: **confidentiality, integrity, and availability**



Attacks Threatening Confidentiality

Snooping

- Refers to **unauthorized access or interception** of data
- Occurs when confidential information is **captured during transmission**
- Example: intercepting a file sent over the Internet
- Prevented by making data **unreadable using encryption (encipherment)**

Traffic Analysis

- Occurs even when data is **encrypted**
- Attacker studies **communication patterns**, not the content
- Can identify **sender/receiver addresses** and timing of messages
- Helps attacker **infer the nature of the transaction**

Attacks Threatening Integrity

The integrity of data can be threatened by several kinds of attacks: modification, masquerading, replaying, and repudiation.

1. Modification Attack

- Attacker intercepts or accesses information and **alters the data** for personal benefit.
  - Example:A customer sends a transaction request to a bank.
  - Attacker intercepts the message and **changes the transaction type or amount**.

In some cases, the attacker may:

Delete the message.

Delay the message to disrupt the system or gain advantage.

Threatens **data integrity**.

Attacks Threatening Integrity

2. Masquerading (Spoofing)

•Attacker **impersonates another legitimate entity**.

Example 1:Attacker steals a customer's **bank card and PIN** and acts as that customer.

Example 2:Attacker pretends to be the **bank server**.

•User unknowingly provides sensitive information.

Threatens **authentication and confidentiality**

Attacks Threatening Integrity

3. Replaying Attack

Attacker **captures a valid message** and retransmits it later.

Example: A user sends a payment request to a bank.

Attacker records the message and **replays it** to receive multiple payments.

Message contents are not altered—only **resent**

Threatens **integrity and availability**.

Attacks Threatening Integrity

4. Repudiation

Performed by a **legitimate participant** (sender or receiver).
Sender may:Deny having **sent** the message.
Receiver may:Deny having **received** the message.

Common in electronic transactions without proper proof.

Threatens n**on-repudiation**.

Attacks Threatening Availability

Denial of Service : Aims to **slow down or completely disrupt** system services.

Common attack strategies:

1.Sending a **large number of fake (bogus) requests** to overload and crash a server.

2.Intercepting and deleting server responses, making clients think the server is down.

3.Blocking or intercepting client requests, causing repeated retransmissions that overload the system.

Affects **availability** of the system.

Passive attacks

Attacker's goal is to **obtain information only**.

- **No modification of data** and no disruption to system operation.
- System continues to function normally.

- May harm the **sender or receiver** by leaking confidential information.
- Includes attacks that threaten **confidentiality**:
  - Snooping
  - Traffic analysis
- Difficult to detect because **no visible system damage** occurs.
- Can be **prevented using data encipherment (encryption)**.

Active Attacks

- An active attack may change the data or harm the system.
- Attacks that threaten the integrity and availability are active attacks.
- Active attacks are normally easier to detect than to prevent, because an attacker can launch them in a variety of ways

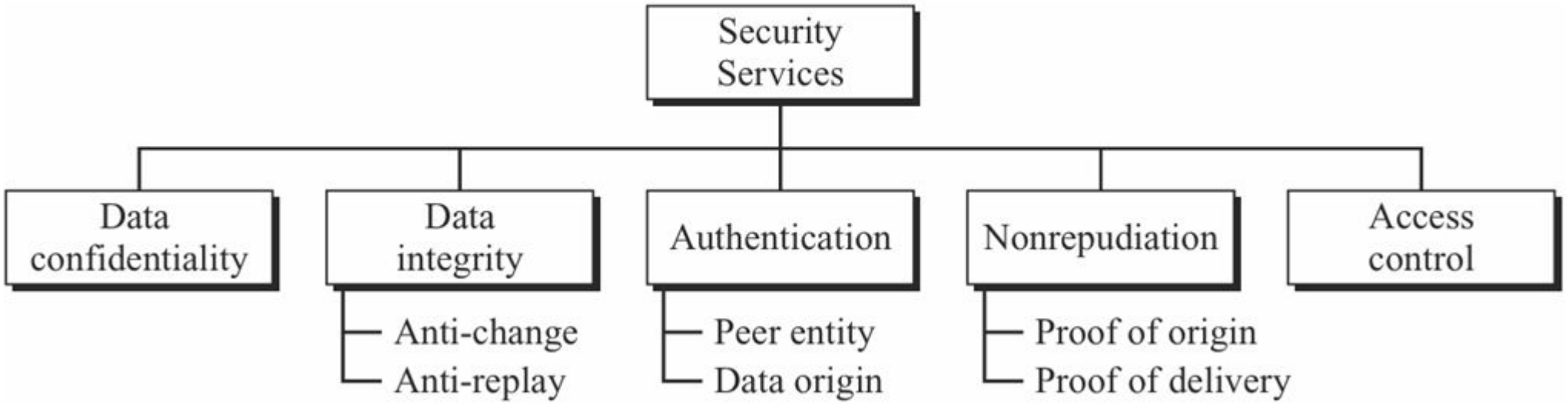
Passive Versus Active Attacks

Attacks	*Passive/Active*	*Threatening*
Snooping Traffic analysis	Passive	Confidentiality
Modification Masquerading Replaying Repudiation	Active	Integrity
Denial of service	Active	Availability

SERVICES AND MECHANISMS

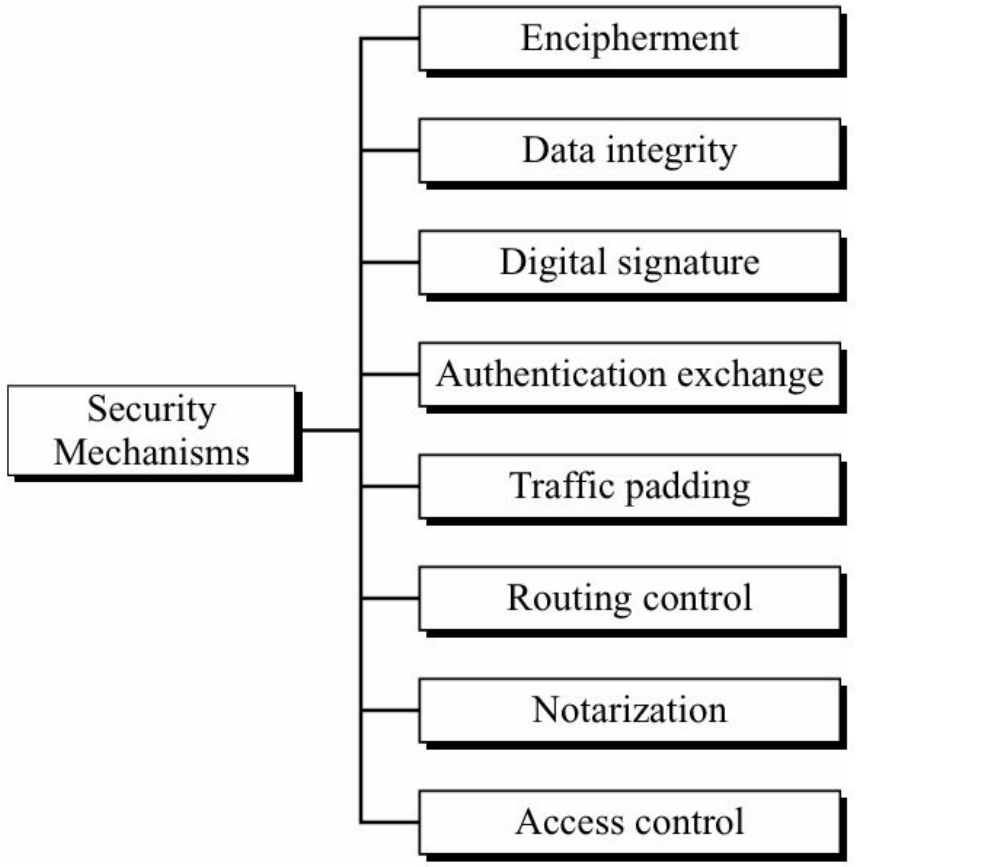
The International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) provides some security services and some mechanisms to implement those services

Security services



Security Mechanisms

Security mechanisms



Relation between Services and Mechanisms

Security Service	*Security Mechanism*
Data confidentiality	Encipherment and routing control
Data integrity	Encipherment, digital signature, data integrity
Authentication	Encipherment, digital signature, authentication exchanges
Nonrepudiation	Digital signature, data integrity, and notarization
Access control	Access control mechanism

TECHNIQUES

- The security mechanisms discussed so far provide only theoretical guidelines;
- Their practical realization relies on concrete techniques, primarily **cryptography**, which offers general-purpose protection,
- And **steganography**, which provides a more specific method of securing information.

Cryptography:

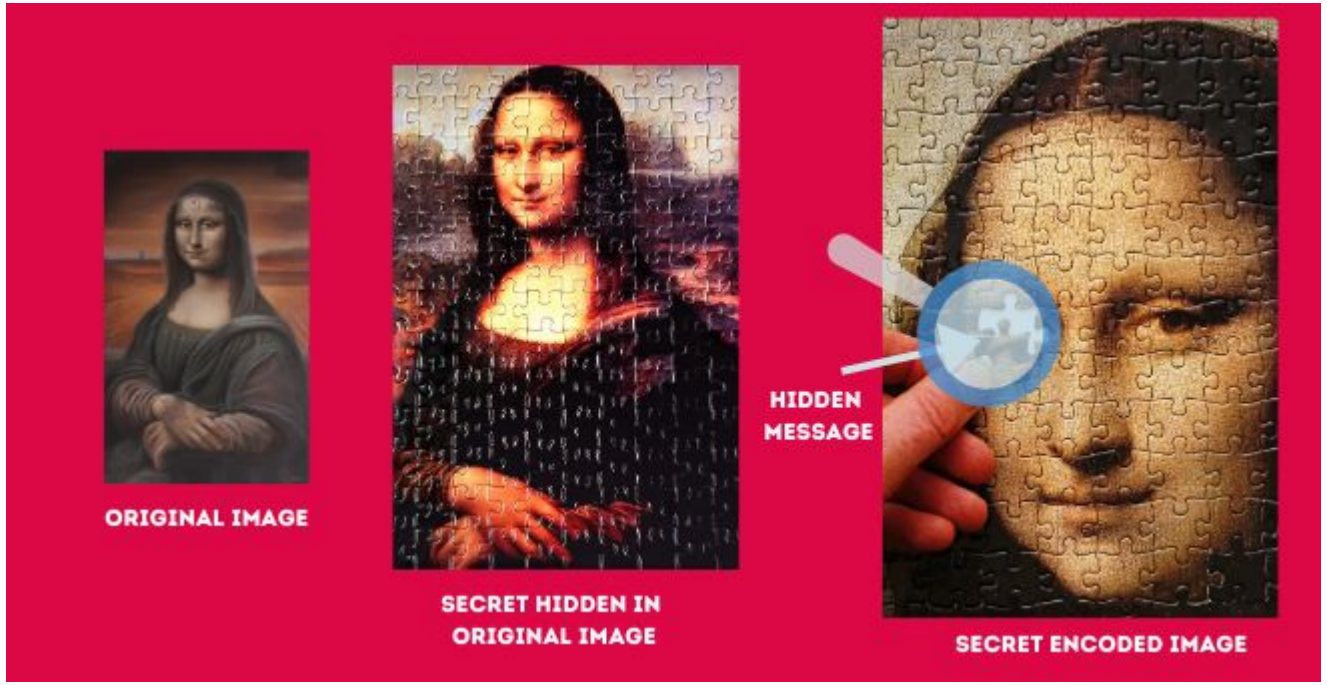
Cryptography, a word with Greek origins, means "secret writing."
However, we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks.

Although in the past cryptography referred only to the encryption and decryption of messages using secret keys, today it is defined as involving three distinct mechanisms:

- symmetric-key encipherment,
- asymmetric-key encipherment
- hashing

Steganography

- ***The word steganography, with origin in Greek, means "covered writing,"*** in contrast with cryptography, which means "secret writing."
- Cryptography means concealing the contents of a message by enciphering;
- Steganography means concealing the message itself by covering it with something else.



CRYPTOGRAPHY

Traditional Symmetric Key Ciphers
Mathematics of Cryptography

Archana M
Department of Computer Applications

Mathematics of Cryptography

Cryptography is based on some specific areas of mathematics, including number theory, linear algebra, and algebraic structures.

INTEGER ARITHMETIC**:**

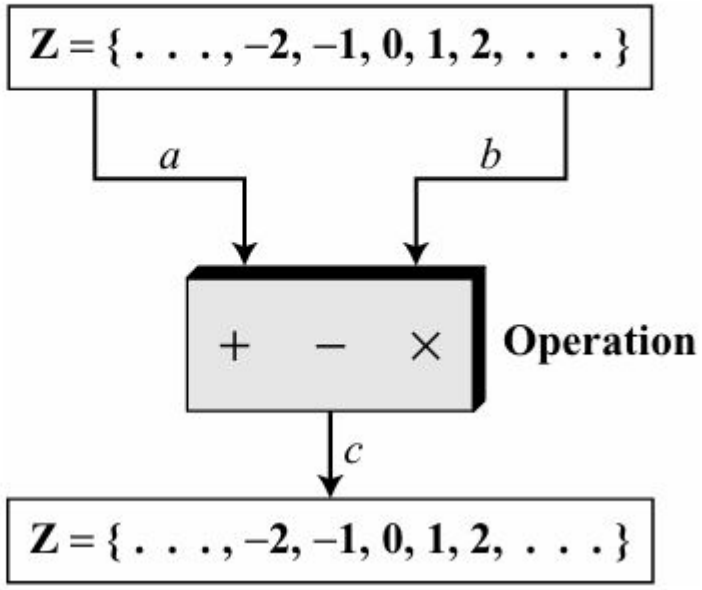
Set of Integers **:** The set of integers, denoted by Z, contains all integral numbers (with no fraction) from negative infinity to positive infinity

$Z = \{ \ldots, -2, -1, 0, 1, 2, \ldots \}$

Binary Operations - A binary operation takes two inputs and creates one output.

Three common binary operations defined for integers are addition, subtraction, and multiplication

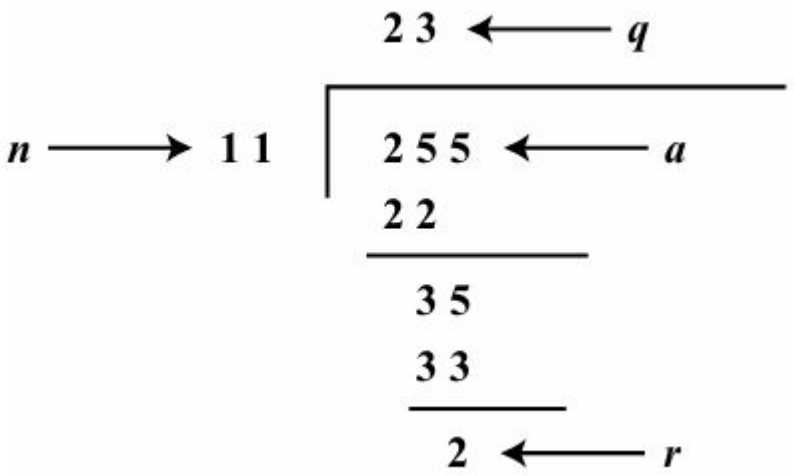
Three binary operations for the set of integers



Integer Division:In integer arithmetic, if we divide a by n, we can get q and r. The relationship between these four integers can be shown as

$$a = q \times n + r$$

Example : Assume that a = 255 and n = 11. We can find q = 23 and r = 2

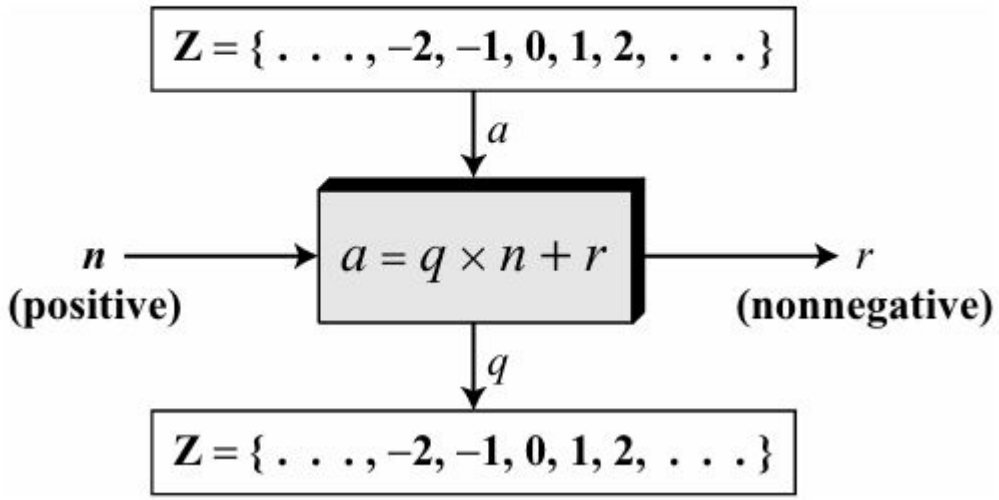


When we use the this division relationship in cryptography, we impose two restrictions:

First, we require that the divisor be a positive integer ($n > 0$).

Second, we require that the remainder be a nonnegative integer ($r \geq 0$).

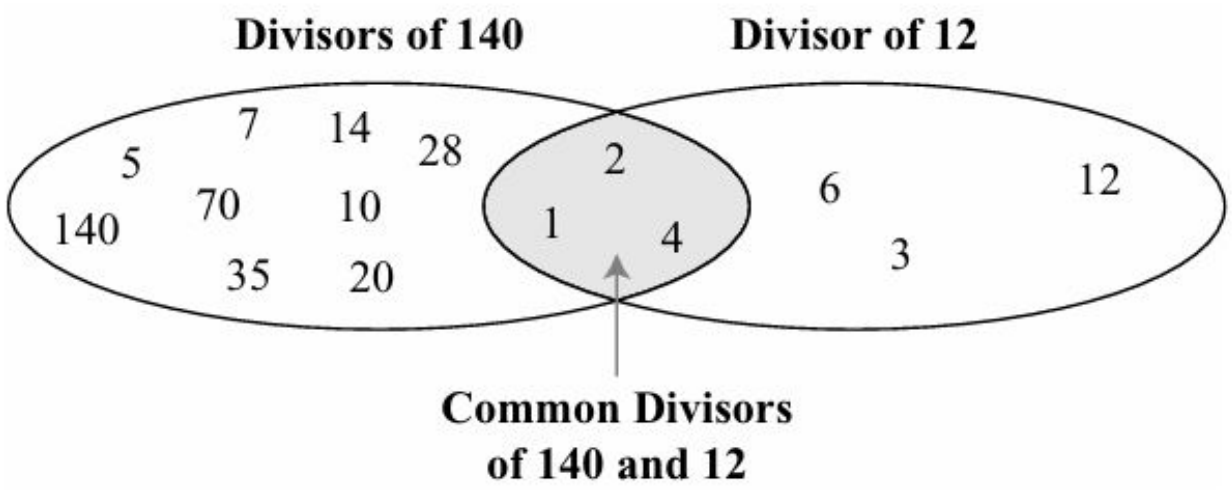
Division algorithm for integers



Greatest Common Divisor

The greatest common divisor of two positive integers is the largest integer that can divide both integers.

Common divisors of two integers



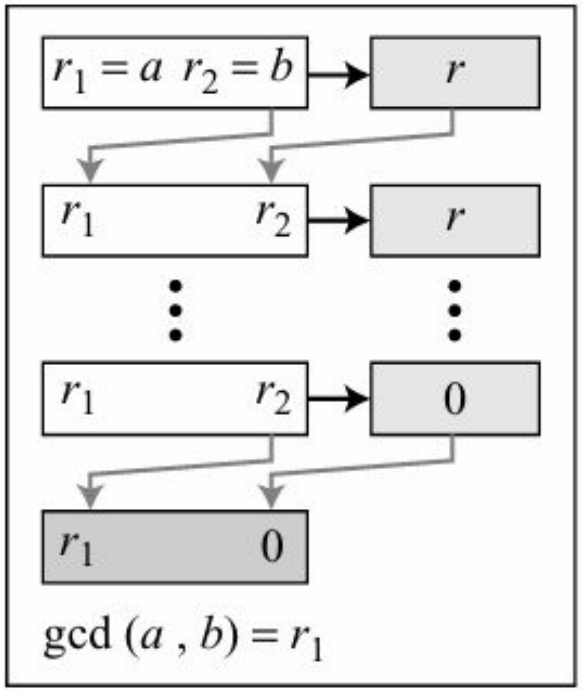
Euclidean Algorithm

- Finding the greatest common divisor (gcd) of two positive integers by listing all common divisors is not practical when the two integers are large.
- So we have **Euclidean algorithm** which is based on two facts
  - Fact 1: gcd (a, 0) = a
  - Fact 2: gcd (a, b) = gcd (b, r), where r is the remainder of dividing a by b

For example, to calculate the gcd (36, 10)

gcd (36, 10) = gcd (10, 6) = gcd (6, 4) = gcd (4, 2) = gcd (2, 0) = 2

Euclidean algorithm



a. Process

```
 r1 ← a; r2 ← b;  (Initialization)
while (r2 > 0)
  {
   q ← r1 / r2;
   r ← r1 – q × r2;
   r1 ← r2; r2 ← r;
  }

   gcd (a, b) ← r1
```

b. Algorithm

Find the greatest common divisor of 2740 and 1760.

We initialize r1 to 2740 and r2 to 1760. We have also shown the value of q in each step. We have gcd (2740, 1760) = 20

q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

Find the greatest common divisor of 25 and 60.

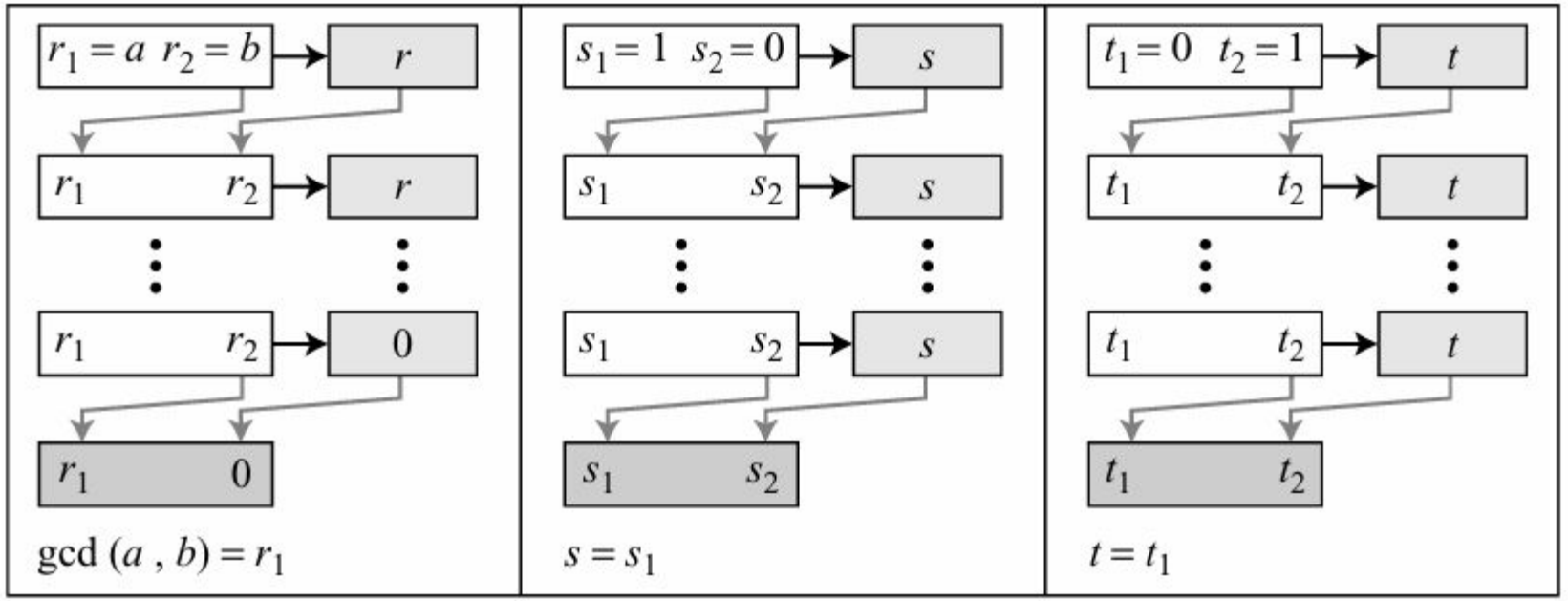
q	r_1	r_2	r
0	25	60	25
2	60	25	10
2	25	10	5
2	10	5	0
	5	0	

The Extended Euclidean Algorithm

Given two integers a and b, we often need to find other two integers, s and t, such that

$$s \times a + t \times b = \gcd(a, b)$$

Extended Euclidean algorithm



a. Process

```
r1 ← a; r2 ← b;
s1 ← 1; s2 ← 0;          (Initialization)
t1 ← 0; t2 ← 1;
while (r2 > 0)
 {
  q ← r1 / r2;
    r  ← r1 - q × r2;
    r1 ← r2;   r2 ← r;          (Updating r's)
    s  ← s1 - q × s2;
    s1 ← s2;   s2 ← s;          (Updating s's)
    t  ← t1 - q × t2;
    t1 ← t2;   t2 ← t;          (Updating t's)
 }
  gcd (a , b) ← r1;   s ← s1;   t ← t1
```

b. Algorithm

Example : Given a = 161 and b = 28, find gcd (a, b) and the values of s and t

Solution :

r= r1 − q × r2 s = s1 − q × s2 t = t1 − q × t2

We use a table to follow the algorithm.

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	−5
1	28	21	7	0	1	−1	1	−5	6
3	21	7	0	1	−1	4	−5	6	−23
	7	0		**−1**	4		**6**	−23	

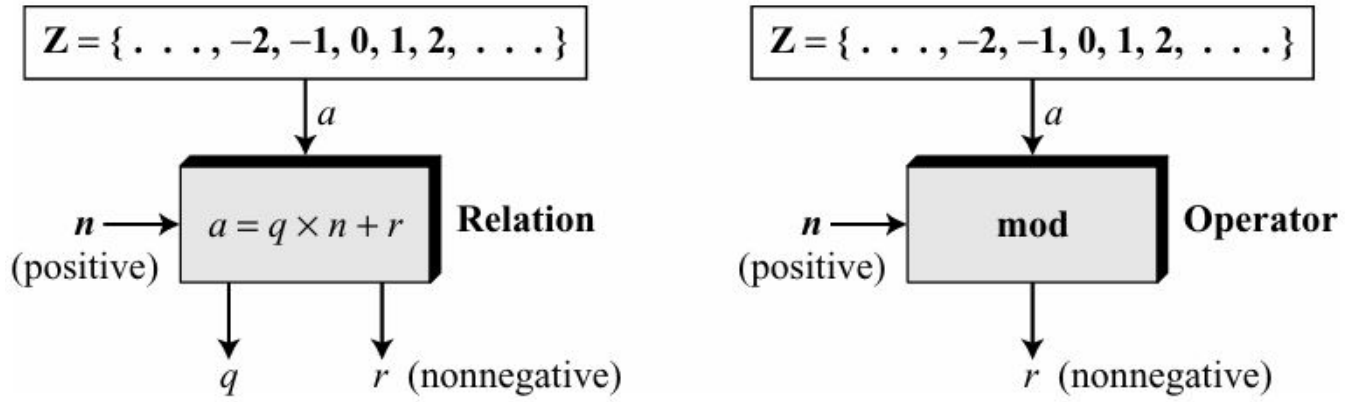
We get gcd (161, 28) = 7, $s = -1$ and $t = 6$. The answers can be tested because we have

$$(-1) \times 161 + 6 \times 28 = 7$$

MODULAR ARITHMETIC

- The division relationship ($a = q \times n + r$) discussed in the previous section has two inputs (a and n) and two outputs (q and r).
- In modular arithmetic, we are interested in only one of the outputs, the remainder r.
- Modulo Operator **:** The above-mentioned binary operator is called the modulo operator and is shown as mod. The second input (n) is called the modulus. The output r is called the residue

Division relation and modulo operator



Find the result of the following operations:

a. $27 \bmod 5$
b. $36 \bmod 12$
c. $-18 \bmod 14$
d. $-7 \bmod 10$

Solution

We are looking for the residue r. We can divide the a by n and find q and r.

We can then disregard q and keep r. a.

1. Dividing 27 by 5 results in r = 2. This means that 27 mod 5 = 2.
2. Dividing 36 by 12 results in r = 0. This means that 36 mod 12 = 0.
3. Dividing −18 by 14 results in r = −4. However, we need to add the modulus (14) to make it nonnegative. We have r = −4 + 14 = 10. This means that −18 mod 14 = 10.
4. Dividing −7 by 10 results in r = −7. After adding the modulus to −7, we have r = 3. This means that −7 mod 10 = 3

Set of Residues: Zn

The result of the modulo operation with modulus n is always an integer between 0 and n − 1

Modulo operation creates a set, which in modular arithmetic is referred to as the set of least residues modulo n, or Zn

For Example:

$$Z_n = \{ 0, 1, 2, 3, \ldots, (n-1) \}$$

$Z_2 = \{ 0, 1 \}$ $Z_6 = \{ 0, 1, 2, 3, 4, 5 \}$ $Z_{11} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$

Congruence

To show that two integers are congruent, we use the congruence operator (≡)

$2 \equiv 12 \pmod{10}$	$13 \equiv 23 \pmod{10}$	$34 \equiv 24 \pmod{10}$	$-8 \equiv 12 \pmod{10}$
$3 \equiv 8 \pmod{5}$	$8 \equiv 13 \pmod{5}$	$23 \equiv 33 \pmod{5}$	$-8 \equiv 2 \pmod{5}$

Perform the following operations (the inputs come from Z_n)

a. Add 7 to 14 in Z_{15} .

b. Subtract 11 from 7 in Z_{15}

c. Multiply 11 by 7 in Z_{20}

Solution

The following shows the two steps involved in each case:

$$(14 + 7) \bmod 15 \rightarrow (21) \bmod 15 = 6$$
$$(7 - 11) \bmod 13 \rightarrow (-4) \bmod 13 = 9$$
$$(7 \times 11) \bmod 20 \rightarrow (77) \bmod 20 = 17$$

CRYPTOGRAPHY

Traditional Symmetric Key Ciphers
Mathematics of Cryptography

Archana M
Department of Computer Applications

Mathematics of Cryptography

Cryptography is based on some specific areas of mathematics, including number theory, linear algebra, and algebraic structures.

INTEGER ARITHMETIC**:**

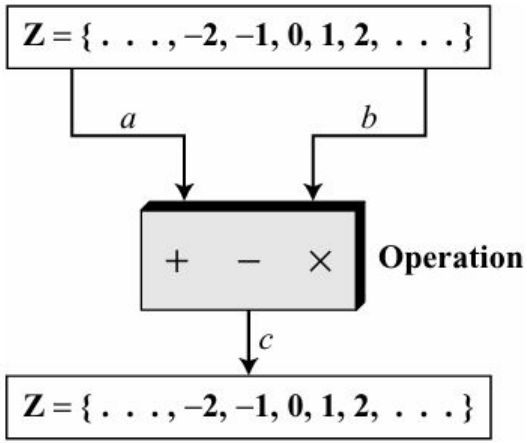
Set of Integers **:** The set of integers, denoted by Z, contains all integral numbers (with no fraction) from negative infinity to positive infinity

$Z = \{ \ldots, -2, -1, 0, 1, 2, \ldots \}$

Binary Operations - A binary operation takes two inputs and creates one output.

Three common binary operations defined for integers are addition, subtraction, and multiplication

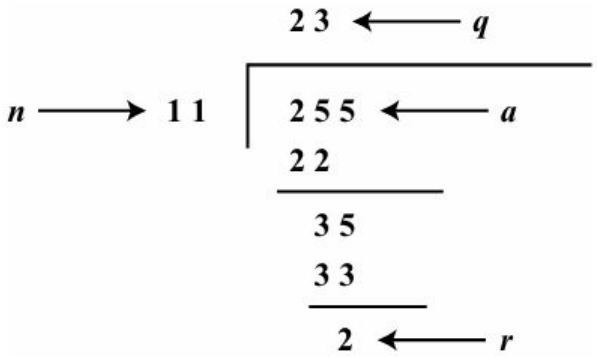
Three binary operations for the set of integers



Integer Division:In integer arithmetic, if we divide a by n, we can get q and r. The relationship between these four integers can be shown as

$$a = q \times n + r$$

Example : Assume that a = 255 and n = 11. We can find q = 23 and r = 2

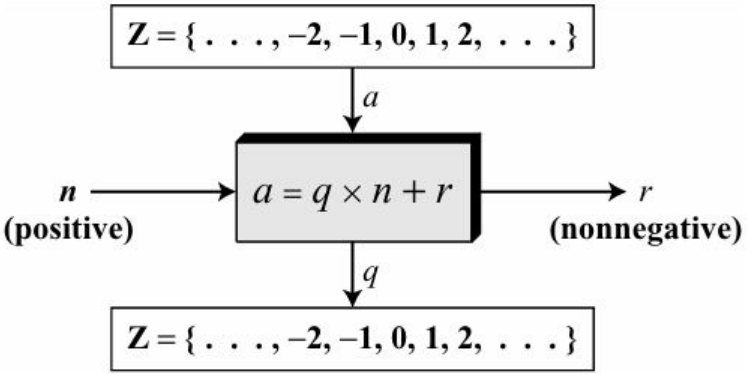


When we use the this division relationship in cryptography, we impose two restrictions:

First, we require that the divisor be a positive integer ($n > 0$).

Second, we require that the remainder be a nonnegative integer ($r \geq 0$).

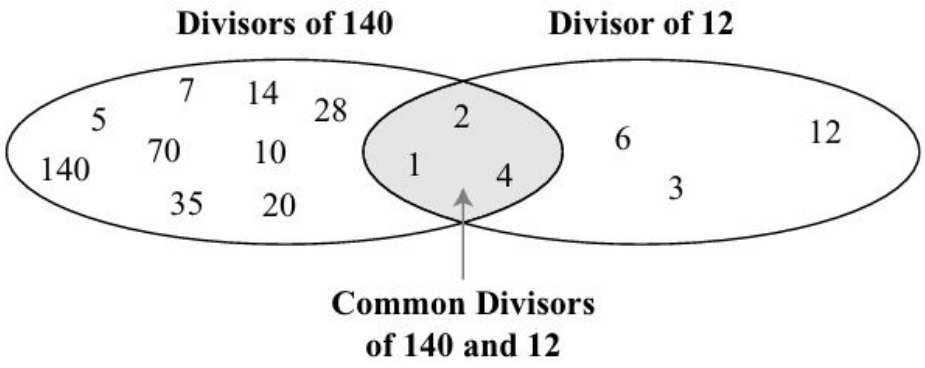
Division algorithm for integers



Greatest Common Divisor

The greatest common divisor of two positive integers is the largest integer that can divide both integers.

Common divisors of two integers



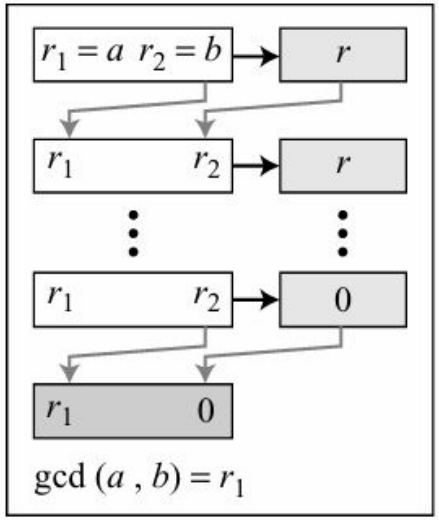
Euclidean Algorithm

- Finding the greatest common divisor (gcd) of two positive integers by listing all common divisors is not practical when the two integers are large.
- So we have **Euclidean algorithm** which is based on two facts
  - Fact 1: gcd (a, 0) = a
  - Fact 2: gcd (a, b) = gcd (b, r), where r is the remainder of dividing a by b

For example, to calculate the gcd (36, 10)

gcd (36, 10) = gcd (10, 6) = gcd (6, 4) = gcd (4, 2) = gcd (2, 0) = 2

Euclidean algorithm



a. Process

```
r1 ← a; r2 ← b;  (Initialization)
while (r2 > 0)
 {
  q ← r1 / r2;
  r ← r1 − q × r2;
  r1 ← r2; r2 ← r;
 }

  gcd (a, b) ← r1
```

b. Algorithm

Find the greatest common divisor of 2740 and 1760.

We initialize r1 to 2740 and r2 to 1760. We have also shown the value of q in each step. We have gcd (2740, 1760) = 20

q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

Find the greatest common divisor of 25 and 60.

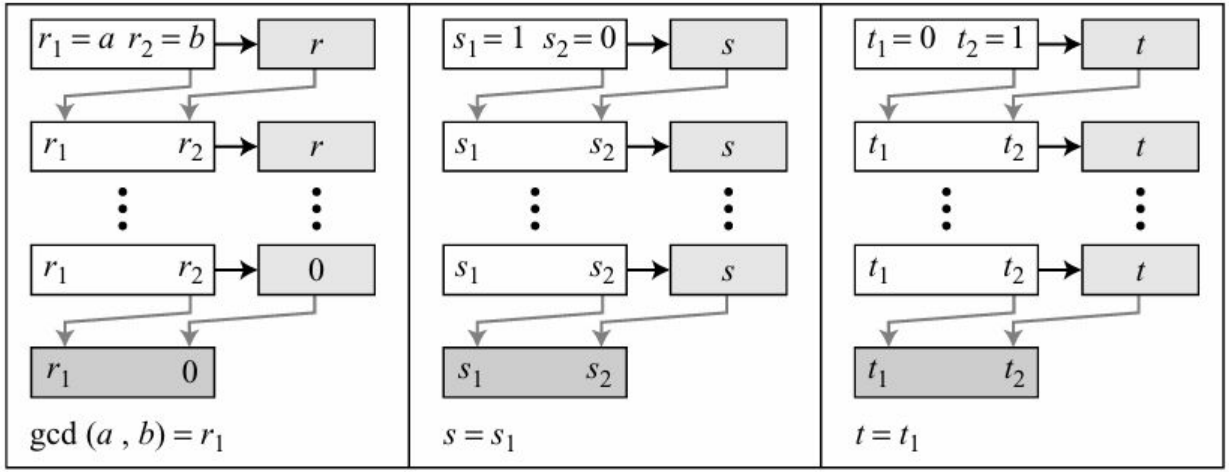
q	r_1	r_2	r
0	25	60	25
2	60	25	10
2	25	10	5
2	10	5	0
	5	0	

The Extended Euclidean Algorithm

Given two integers a and b, we often need to find other two integers, s and t, such that

$$s \times a + t \times b = \gcd(a, b)$$

Extended Euclidean algorithm



a. Process

```
r1 ← a; r2 ← b;
s1 ← 1; s2 ← 0;          (Initialization)
t1 ← 0; t2 ← 1;
while (r2 > 0)
 {
  q ← r1 / r2;
    r  ← r1 – q × r2;
    r1 ← r2;   r2 ← r;          (Updating r's)
    s  ← s1 – q × s2;
    s1 ← s2;   s2 ← s;          (Updating s's)
    t  ← t1 – q × t2;
    t1 ← t2;   t2 ← t;          (Updating t's)
 }
  gcd (a , b) ← r1;   s ← s1;   t ← t1
```

b. Algorithm

Example : Given a = 161 and b = 28, find gcd (a, b) and the values of s and t

Solution **:**

r= r1 − q × r2 s = s1 − q × s2 t = t1 − q × t2

We use a table to follow the algorithm.

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	−5
1	28	21	7	0	1	−1	1	−5	6
3	21	7	0	1	−1	4	−5	6	−23
	7	0		**−1**	4		**6**	−23	

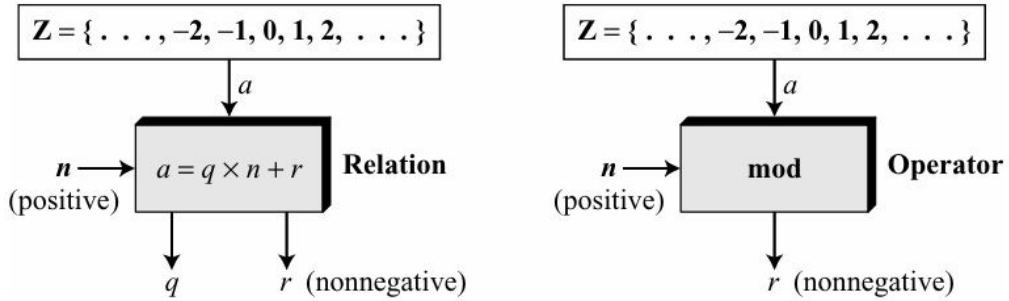
We get gcd (161, 28) = 7, $s = -1$ and $t = 6$. The answers can be tested because we have

$$(-1) \times 161 + 6 \times 28 = 7$$

MODULAR ARITHMETIC

- The division relationship (a = q × n + r) discussed in the previous section has two inputs (a and n) and two outputs (q and r).
- In modular arithmetic, we are interested in only one of the outputs, the remainder r.
- Modulo Operator **:** The above-mentioned binary operator is called the modulo operator and is shown as mod. The second input (n) is called the modulus. The output r is called the residue

Division relation and modulo operator



Find the result of the following operations:

a. $27 \bmod 5$
b. $36 \bmod 12$
c. $-18 \bmod 14$
d. $-7 \bmod 10$

Solution

We are looking for the residue r. We can divide the a by n and find q and r.

We can then disregard q and keep r. a.

1. Dividing 27 by 5 results in r = 2. This means that 27 mod 5 = 2.
2. Dividing 36 by 12 results in r = 0. This means that 36 mod 12 = 0.
3. Dividing −18 by 14 results in r = −4. However, we need to add the modulus (14) to make it nonnegative. We have r = −4 + 14 = 10. This means that −18 mod 14 = 10.
4. Dividing −7 by 10 results in r = −7. After adding the modulus to −7, we have r = 3. This means that −7 mod 10 = 3

Set of Residues: Zn

The result of the modulo operation with modulus n is always an integer between 0 and n − 1

Modulo operation creates a set, which in modular arithmetic is referred to as the set of least residues modulo n, or Zn

For Example:

$$Z_n = \{ 0, 1, 2, 3, \ldots, (n-1) \}$$

$Z_2 = \{ 0, 1 \}$ $Z_6 = \{ 0, 1, 2, 3, 4, 5 \}$ $Z_{11} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$

Congruence

To show that two integers are congruent, we use the congruence operator (≡)

$2 \equiv 12 \pmod{10}$	$13 \equiv 23 \pmod{10}$	$34 \equiv 24 \pmod{10}$	$-8 \equiv 12 \pmod{10}$
$3 \equiv 8 \pmod{5}$	$8 \equiv 13 \pmod{5}$	$23 \equiv 33 \pmod{5}$	$-8 \equiv 2 \pmod{5}$

Perform the following operations (the inputs come from Z_n)

a. Add 7 to 14 in Z_{15} .

b. Subtract 11 from 7 in Z_{15}

c. Multiply 11 by 7 in Z_{20}

Solution

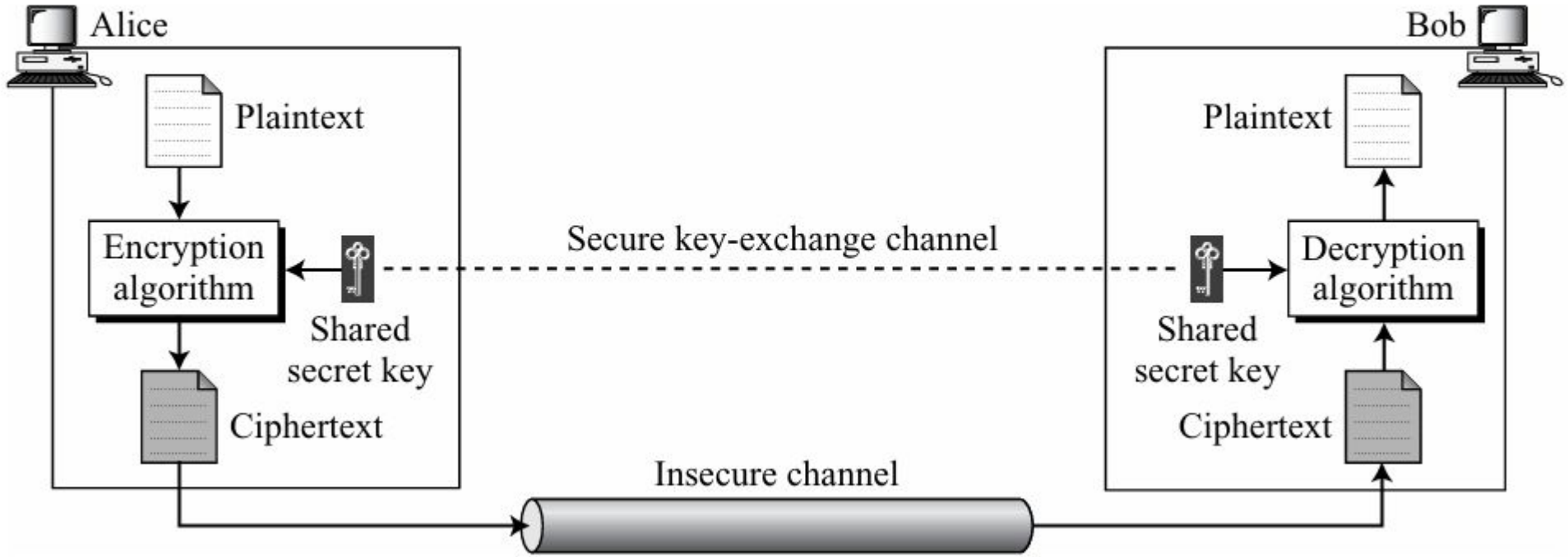
The following shows the two steps involved in each case:

$$(14 + 7) \bmod 15 \rightarrow (21) \bmod 15 = 6$$
$$(7 - 11) \bmod 13 \rightarrow (-4) \bmod 13 = 9$$
$$(7 \times 11) \bmod 20 \rightarrow (77) \bmod 20 = 17$$

Traditional Symmetric-Key Ciphers

- These ciphers are not used today but, they are simpler than modern ciphers and easier to understand.
- They show the basic foundation of cryptography and encipherment.
- They provide the rationale for using modern ciphers, because the traditional ciphers can be easily attacked using a computer

General idea of symmetric-key cipher



- The original message from Alice to Bob is called plaintext.
- The message that is sent through the channel is called the ciphertext.
- To create the ciphertext from the plaintext, Alice uses an encryption algorithm and a shared secret key
- To create the plaintext from ciphertext, Bob uses a decryption algorithm and the same secret key
- We refer to encryption and decryption algorithms as ciphers.
- A key is a set of values (numbers) that the cipher, as an algorithm, operates on

- Note that the symmetric-key encipherment uses a single key (the key itself may be a set of values) for both encryption and decryption.
- The encryption and decryption algorithms are inverses of each other.
- If P is the plaintext, C is the ciphertext, and K is the key, the encryption algorithm $E_k(x)$ creates the ciphertext from the plaintext; the decryption algorithm $D_k(x)$ creates the ciphertext from the plaintext;
- We assume that $E_k(x)$ and $D_k(x)$ are inverses of each other: they cancel the effect of each other if they are applied one after the other on the same input.

Encryption: $C = E_k(P)$ Decryption: $P = D_k(C)$

In which, $D_k(E_k(x)) = E_k(D_k(x)) = x$

We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates P_1 ; we prove that $P_1 = P$

Alice: $C = E_k(P)$ **Bob:** $P_1 = D_k(C) = D_k(E_k(P)) = P$

Symmetric-key encipherment as locking and unlocking with the same key

Kerckhoff's Principle

- Kerckhoff's principle states that one should always assume that the adversary, Eve, knows the encryption/decryption algorithm.
- The resistance of the cipher to attack must be based only on the secrecy of the key.
- In other words, guessing the key should be so difficult that there is no need to hide the encryption/decryption algorithm.
- The key domain for each algorithm, how ever, is so large that it makes it difficult for the adversary to find the key

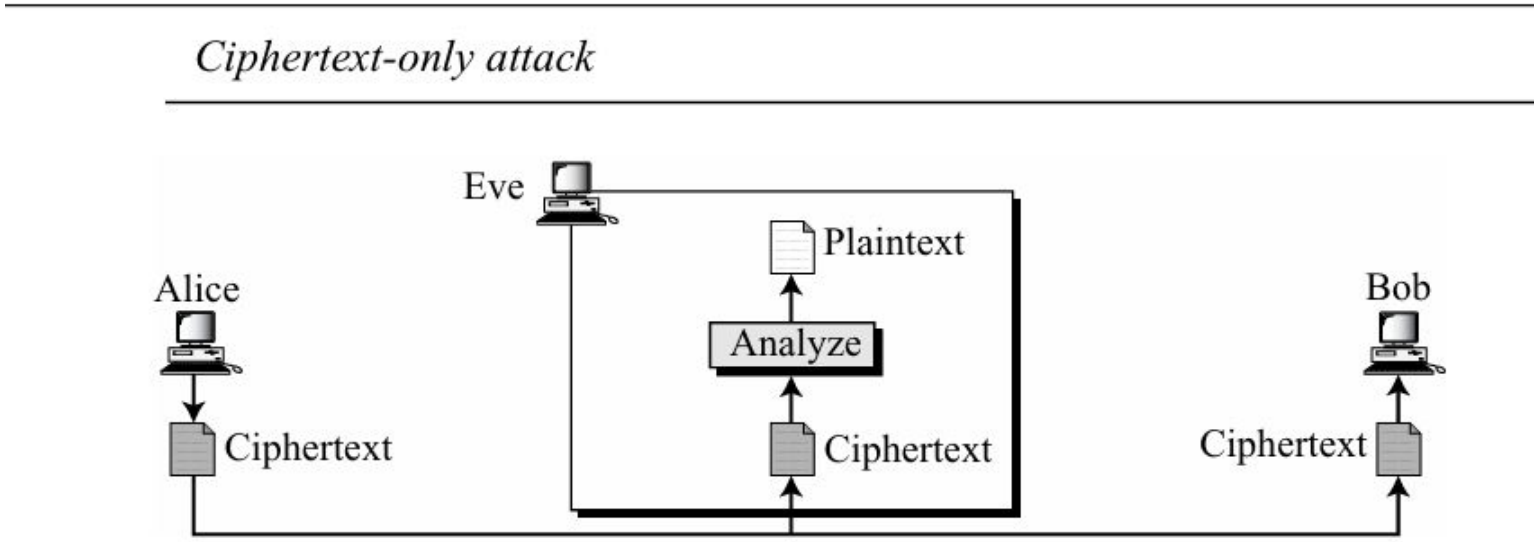
CRYPTOGRAPHY

Traditional Symmetric Key Ciphers

Archana M
Department of Computer Applications

Ciphertext-Only Attack

- In a ciphertext-only attack, Eve has access to only some ciphertext.
- She tries to find the corresponding key and the plaintext.
- The assumption is that Eve knows the algorithm and can intercept the ciphertext.
- The ciphertext-only attack is the most probable one because Eve needs only the ciphertext for this attack.



Various methods can be used in ciphertext-only attack

1. Brute-Force Attack
2. Statistical Attack
3. Pattern Attack

Brute-Force Attack

- In the brute-force method or exhaustive-key-search method, Eve tries to use all possible keys.
- We assume that Eve knows the algorithm and knows the key domain (the list of all possible keys)
- Using the intercepted cipher, Eve decrypts the ciphertext with every possible key until the plaintext makes sense.

Example:

Eve has intercepted the ciphertext “UVACLYFZLJBYL”.

Show how she can use a brute-force attack to break the cipher.

Solution

Eve tries keys from 1 to 7. With a key of 7, the plaintext is “not very secure”, which makes sense.

Ciphertext: UVACLYFZLJBYL

K = 1	→	**Plaintext:** tuzbkxeykiaxk
K = 2	→	**Plaintext:** styajwdxjhzwj
K = 3	→	**Plaintext:** rsxzivcwigyvi
K = 4	→	**Plaintext:** qrwyhubvhfxuh
K = 5	→	**Plaintext:** pqvxgtaugewtg
K = 6	→	**Plaintext:** opuwfsztfdvsf
K = 7	→	**Plaintext:** notverysecure

Statistical Attack

- Statistical Attacks are possible if the adversary has a long ciphertext.
- The adversary can use the frequency of occurrence of characters for a particular language.

Frequency of occurrence of letters in an English text

Letter	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

Statistical Attack

Example: Eve has intercepted the following ciphertext. Using a statistical attack, find the plaintext.

XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEVWMXMWASVX-LQSVILY-
VVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW

Solution

- When Eve tabulates the frequency of letters in this ciphertext, she gets: I =14, V =13, S =12, and so on.
- The most common character is I with 14 occurrences.
- This shows that character I in the ciphertext probably corresponds to the character e in plaintext.
- This means key = 4. Eve deciphers the text to get

Solution

the house is now for sale for four million dollars it is worth more hurry before the seller receives more offers

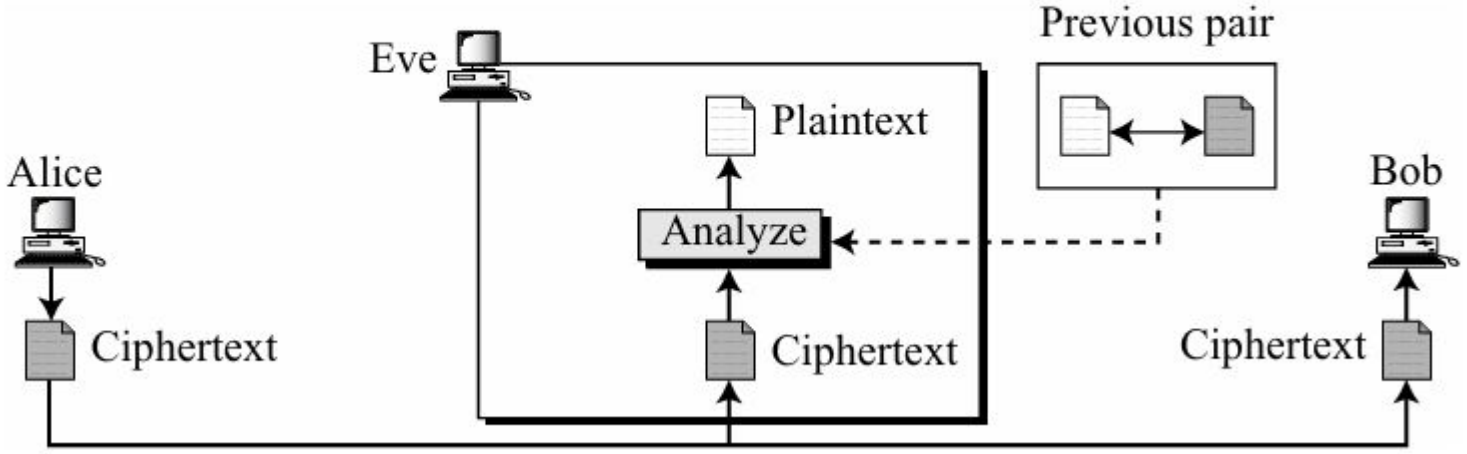
Pattern Attack

- Some ciphers may hide the characteristics of the language, but may create some patterns in the ciphertext.
- A cryptanalyst may use a pattern attack to break the cipher.
- Therefore, it is important to use ciphers that make the ciphertext look as random as possible.

Known-Plaintext Attack

- In a known-plaintext attack, Eve has access to some plaintext/ciphertext pairs in addition to the intercepted ciphertext that she wants to break

Known-plaintext attack



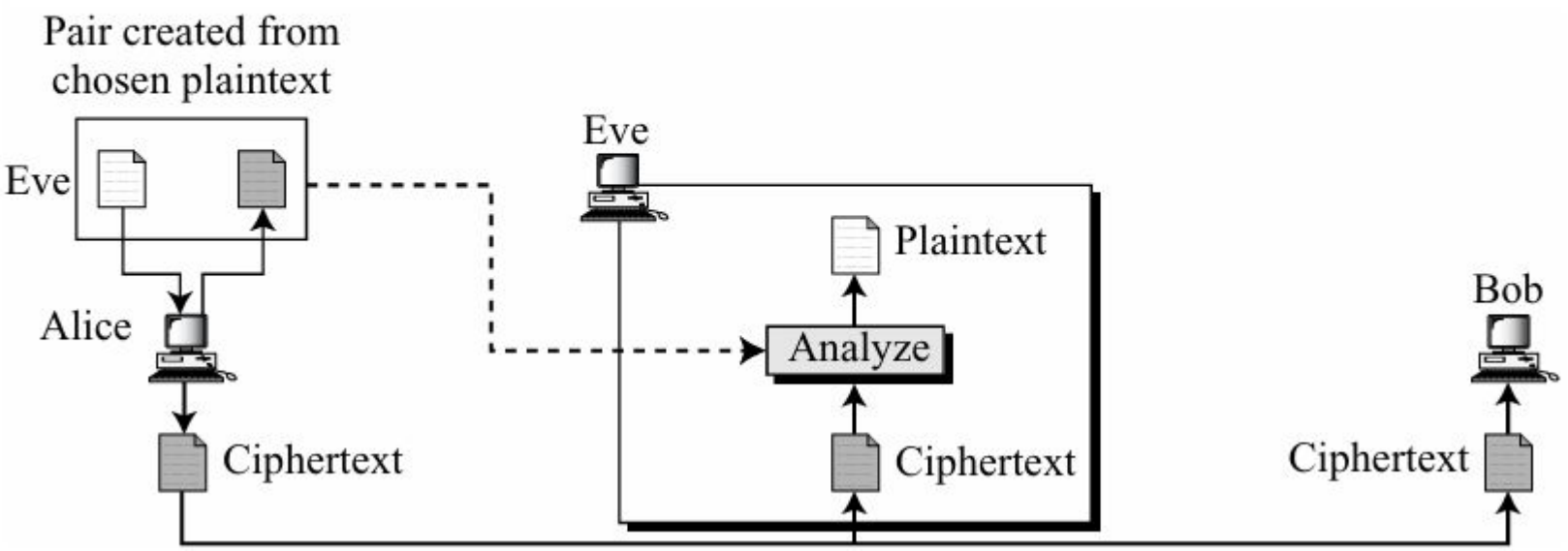
- For example, Alice has sent a secret message to Bob, but she has later made the contents of the message public.
- Eve has kept both the ciphertext and the plaintext to use them to break the next secret message from Alice to Bob, ***assuming that Alice has not changed her key.***
- Eve uses the relationship between the previous pair to analyze the current ciphertext.
  - The same methods used in a ciphertext-only attack can be applied***.***

Chosen-Plaintext Attack

The chosen-plaintext attack is similar to the known-plaintext attack, but the plaintext/ ciphertext pairs have been chosen by the attacker herself

- if Eve has access to Alice's computer.
- She can choose some plaintext and intercept the created ciphertext

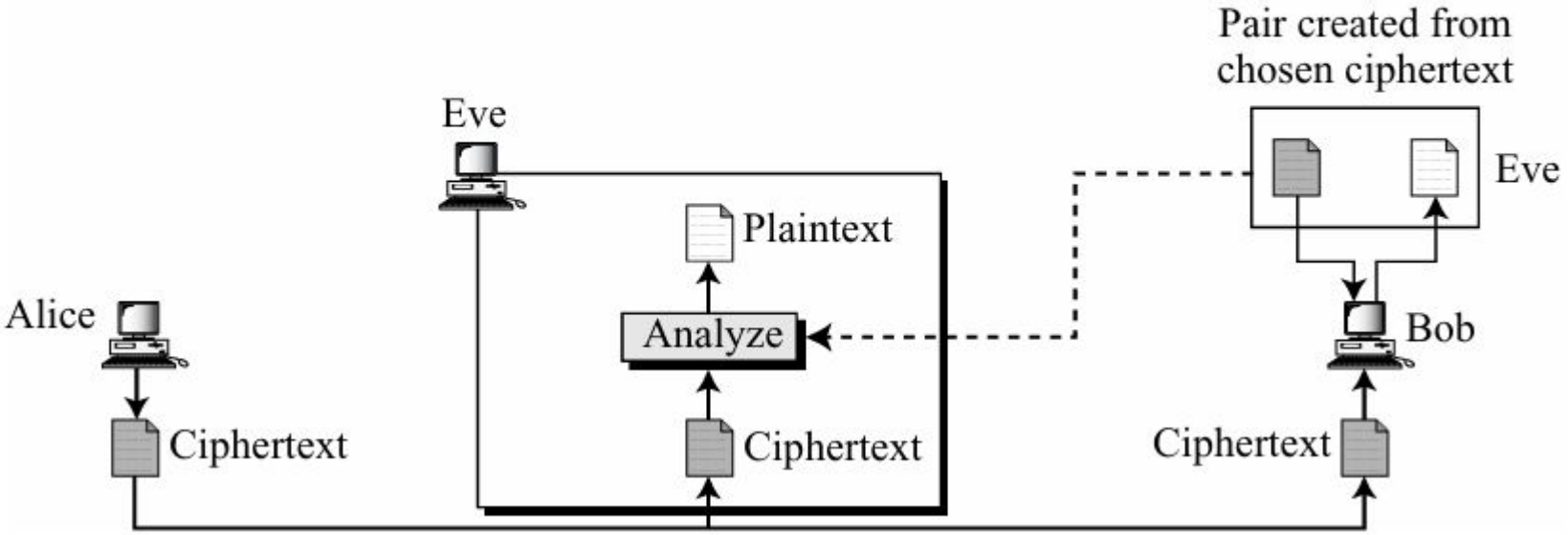
Chosen-plaintext attack



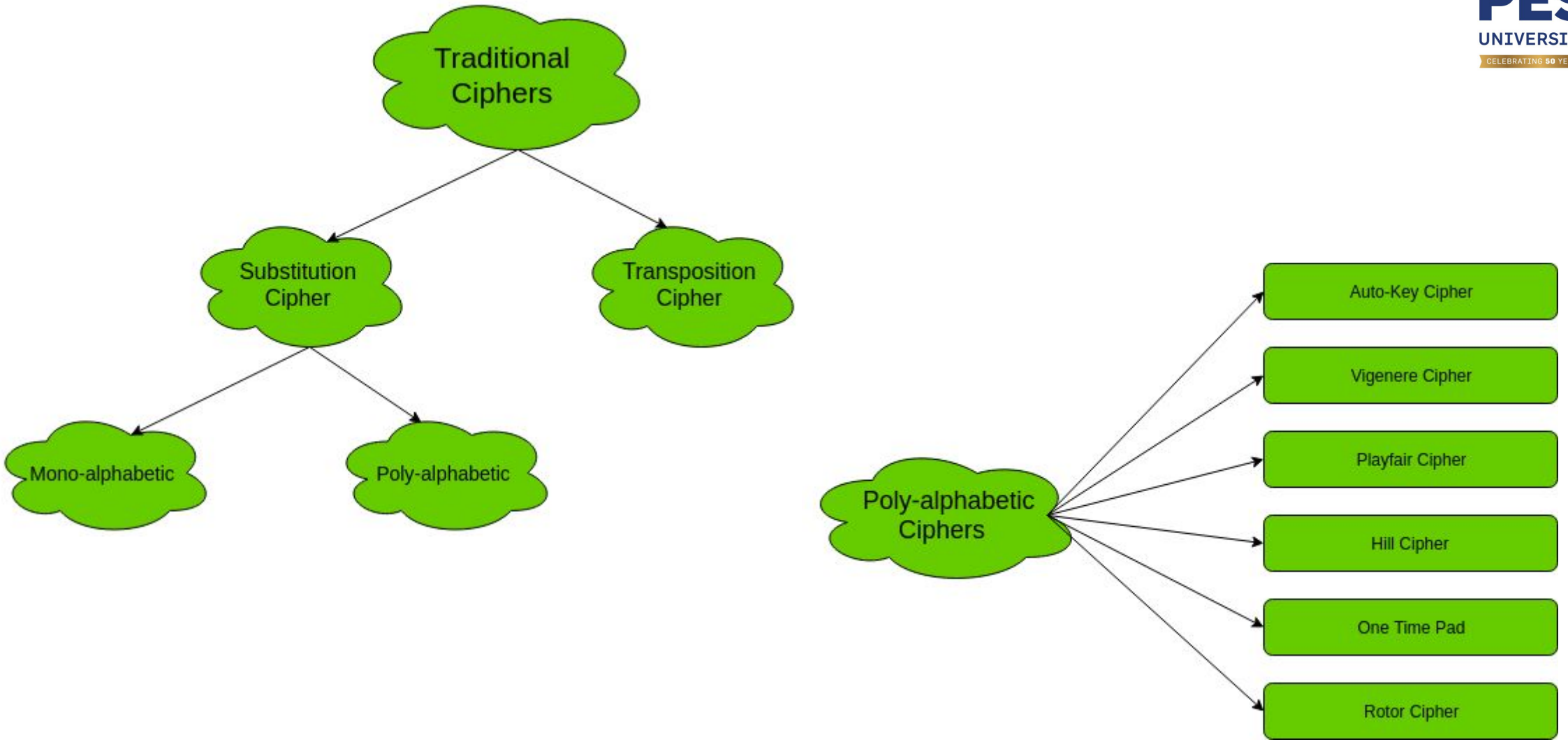
Chosen-Ciphertext Attack

- The chosen-ciphertext attack is similar to the chosen-plaintext attack, except that Eve chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair.
- This can happen if Eve has access to Bob's computer

Chosen-ciphertext attack

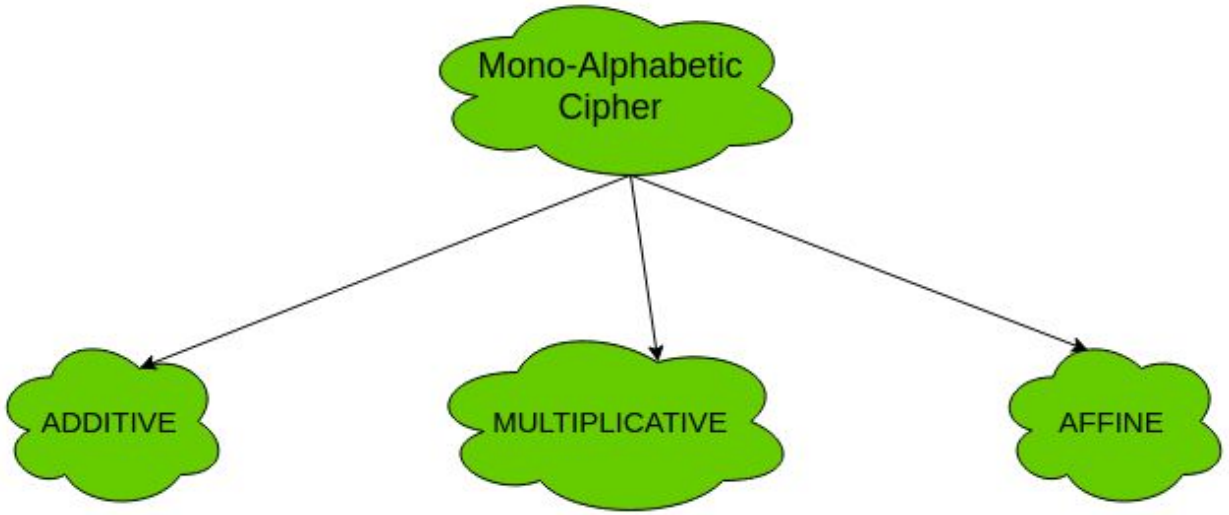


Categories of Traditional Ciphers



Substitution cipher

- A substitution cipher replaces one symbol with another
- If the symbols in the plaintext are alphabetic characters, we replace one character with another.
- For example, we can replace letter A with letter D, and letter T with letter Z. If the symbols are digits (0 to 9), we can replace 3 with 7, and 2 with 6



Substitution Cipher

Monoalphabetic Ciphers

In monoalphabetic substitution, a character (or a symbol) in the plaintext is always changed to the same character (or symbol) in the ciphertext regardless of its position in the text.

For example,

if the algorithm says that letter A in the plaintext is changed to letter D, every letter A is changed to letter D.

In other words, the relationship between letters in the plaintext and the ciphertext is one-to-one

Example 1

Plaintext: hello **Ciphertext:** KHOOR

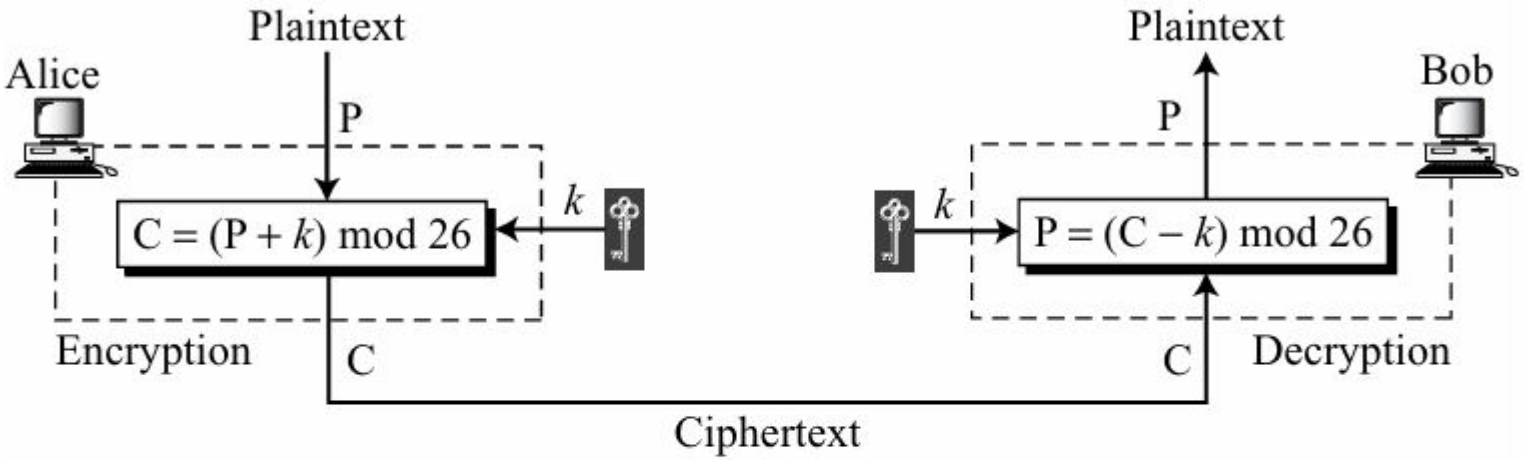
Additive Cipher

- The simplest monoalphabetic cipher is the additive cipher.
- This cipher is also called a shift cipher or Caesar cipher.

Representation of plaintext and ciphertext characters in $\mathbf{Z}_{26}$

Plaintext →	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 →	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

Additive cipher



We can easily prove that the encryption and decryption are inverse of each other because plaintext created by Bob (P_1) is the same as the one sent by Alice (P).

$$P_1 = (C - k) \bmod 26 = (P + k - k) \bmode 26 = P$$

When the cipher is additive, the plaintext, ciphertext, and key are integers in Z_{26}.

Example 1 :

Use the additive cipher with key = 15 to encrypt the message “hello”.

Solution

Plaintext: h → 07	Encryption: (07 + 15) mod 26	Ciphertext: 22 → W
Plaintext: e → 04	Encryption: (04 + 15) mod 26	Ciphertext: 19 → T
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: l → 11	Encryption: (11 + 15) mod 26	Ciphertext: 00 → A
Plaintext: o → 14	Encryption: (14 + 15) mod 26	Ciphertext: 03 → D

The result is “WTAAD”. Note that the cipher is monoalphabetic because two instances of the same plaintext character (l’s) are encrypted as the same character (A).

Now, Use the additive cipher with key = 15 to decrypt the message “WTAAD”.

Solution:

We apply the decryption algorithm to the plaintext character by character:

Ciphertext: W → 22	Decryption: (22 – 15) mod 26	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: (19 – 15) mod 26	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: (00 – 15) mod 26	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: (03 – 15) mod 26	Plaintext: 14 → o

The result is “hello”. Note that the operation is in modulo 26 (see Chapter 2), which means that a negative result needs to be mapped to $\mathbf{Z}_{26}$ (for example −15 becomes 11).

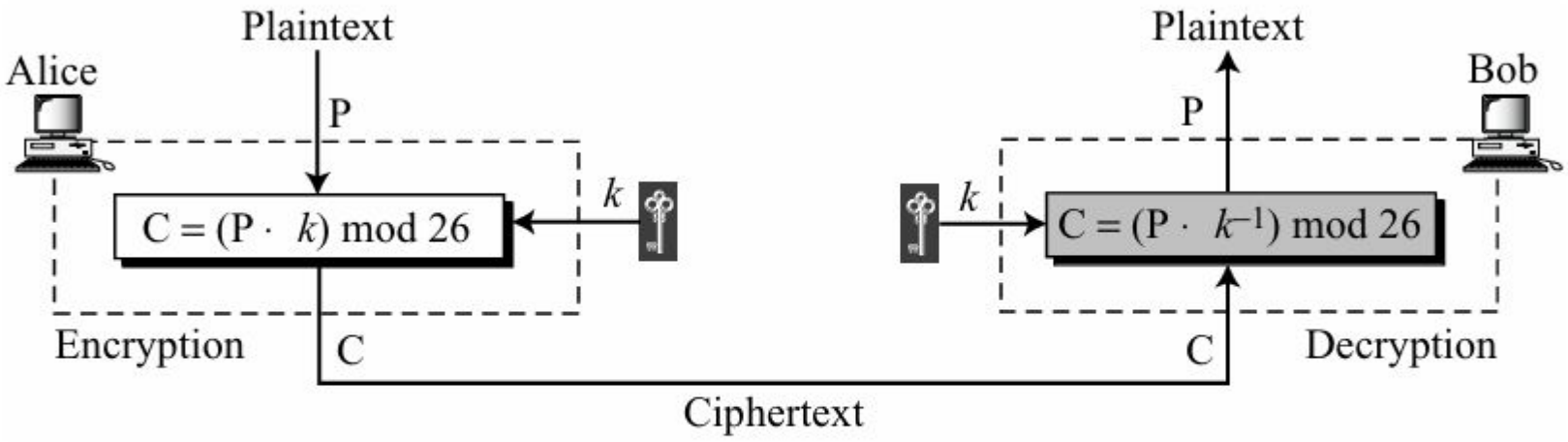
Multiplicative Ciphers

- In a multiplicative cipher, the encryption algorithm specifies multiplication of the plaintext by the key and the decryption algorithm specifies division of the ciphertext by the key

- What is the key domain for any multiplicative cipher
  - The key needs to be in Z_{26}^*,This set has only 12 members: ; the key 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.

Multiplicative Ciphers

Multiplicative cipher



In a multiplicative cipher, the plaintext and ciphertext are integers in Z_{26}; the key is an integer in Z_{26}^*.

Multiplicative Ciphers

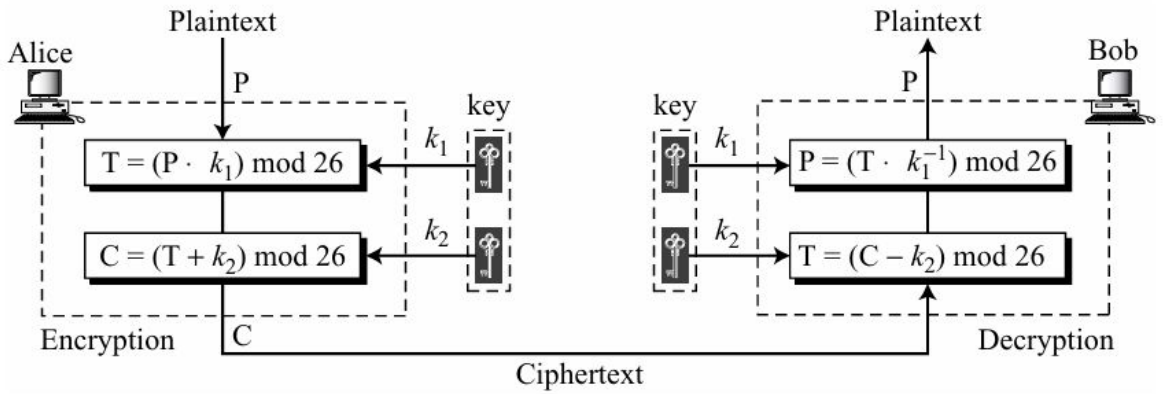
We use a multiplicative cipher to encrypt the message "hello" with a key of 7. The ciphertext is "XCZZU".

Plaintext: h $\rightarrow$ 07	Encryption: $(07 \times 07) \bmod 26$	ciphertext: 23 $\rightarrow$ X
Plaintext: e $\rightarrow$ 04	Encryption: $(04 \times 07) \bmod 26$	ciphertext: 02 $\rightarrow$ C
Plaintext: l $\rightarrow$ 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 $\rightarrow$ Z
Plaintext: l $\rightarrow$ 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 $\rightarrow$ Z
Plaintext: o $\rightarrow$ 14	Encryption: $(14 \times 07) \bmod 26$	ciphertext: 20 $\rightarrow$ U

Affine Cipher

- Combination of the additive and multiplicative ciphers is called the affine cipher.
- The first key is used with the multiplicative cipher; the second key is used with the additive cipher

Affine cipher



In the affine cipher, the relationship between the plaintext P and the ciphertext C is

$$C = (P \times k_1 + k_2) \bmod 26 \qquad P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2

Encryption using Affine Cipher

Use an affine cipher to encrypt the message "hello" with the key pair (7, 2).

Solution

We use 7 for the multiplicative key and 2 for the additive key. We get "ZEBBW".

P: h $\rightarrow$ 07	Encryption: $(07 \times 7 + 2) \bmod 26$	C: 25 $\rightarrow$ Z
P: e $\rightarrow$ 04	Encryption: $(04 \times 7 + 2) \bmod 26$	C: 04 $\rightarrow$ E
P: l $\rightarrow$ 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 $\rightarrow$ B
P: l $\rightarrow$ 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 $\rightarrow$ B
P: o $\rightarrow$ 14	Encryption: $(14 \times 7 + 2) \bmod 26$	C: 22 $\rightarrow$ W

Decryption using Affine Cipher

Use the affine cipher to decrypt the message "ZEBBW" with the key pair (7, 2) in modulus 26.

Solution

Add the additive inverse of $-2 \equiv 24 \pmod{26}$ to the received ciphertext. Then multiply the result by the multiplicative inverse of $7^{-1} \equiv 15 \pmod{26}$ to find the plaintext characters. Because 2 has an additive inverse in $\mathbf{Z}_{26}$ and 7 has a multiplicative inverse in $\mathbf{Z}_{26}^*$, the plaintext is exactly what we used in Example 3.10.

C: Z $\rightarrow$ 25	Decryption: $((25 - 2) \times 7^{-1}) \bmod 26$	P:07 $\rightarrow$ h
C: E $\rightarrow$ 04	Decryption: $((04 - 2) \times 7^{-1}) \bmod 26$	P:04 $\rightarrow$ e
C: B $\rightarrow$ 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P:11 $\rightarrow$ l
C: B $\rightarrow$ 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P:11 $\rightarrow$ l
C: W $\rightarrow$ 22	Decryption: $((22 - 2) \times 7^{-1}) \bmod 26$	P:14 $\rightarrow$ o

Polyalphabetic Ciphers

- In polyalphabetic substitution, each occurrence of a character may have a different substitute.
- The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.
- For example, “a” could be enciphered as “D” in the beginning of the text, but as “N” at the middle
- Polyalphabetic ciphers have the advantage of hiding the letter frequency of the underlying language
- Eve cannot use single-letter frequency statistic to break the ciphertext.

- To create a polyalphabetic cipher, we need to make each ciphertext character dependent on both the corresponding plaintext character and the position of the plain text character in the message.
- we need to have a key stream k = (k 1 , k 2 , k 3 , . . .) in which k i is used to encipher the i th character in the plaintext to create the i th character in the ciphertext

Autokey Cipher

- In this cipher, the key is a stream of subkeys, in which each subkey is used to encrypt the corresponding character in the plaintext.
- The first subkey is a predetermined value secretly agreed upon by Alice and Bob.
- The second subkey is the value of the first plaintext character (between 0 and 25).
- The third subkey is the value of the second plaintext. And so on.

$$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3 \ldots \qquad k = (k_1, P_1, P_2, \ldots)$$

$$\text{Encryption: } C_i = (P_i + k_i) \bmod 26 \qquad \text{Decryption: } P_i = (C_i - k_i) \bmod 26$$

- The name of the cipher, autokey , implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.
- Example : Assume that Alice and Bob agreed to use an autokey cipher with initial key value k 1 = 12
  - Alice wants to send Bob the message **“Attack is today”**
  - Enciphering is done character by character. Each character in the plaintext is first replaced by its integer value

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P’s Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	*12*	*00*	*19*	*19*	*00*	*02*	*10*	*08*	*18*	*19*	*14*	*03*	*00*
C’s Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	**M**	**T**	**M**	**T**	**C**	**M**	**S**	**A**	**L**	**H**	**R**	**D**	**Y**

Playfair Cipher

- Another example of a polyalphabetic cipher is the Playfair cipher used by the British army during World War I
- The secret key in this cipher is made of 25 alphabet letters arranged in a 5 × 5 matrix (letters I and J are considered the same when encrypting).
- Different arrangements of the letters in the matrix can create many different secret keys

An example of a secret key in the Playfair cipher

Secret Key =

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

Playfair Cipher

- Before encryption, if the two letters in a pair are the same, a bogus letter is inserted to separate them.
- After inserting bogus letters, if the number of characters in the plaintext is odd, one extra bogus character is added at the end to make the number of characters even.
- The cipher uses three rules for encryption:
  - a. If the two letters in a pair are located in the same row of the secret key, the corre sponding encrypted character for each letter is the next letter to the right in the same row (with wrapping to the beginning of the row if the plaintext letter is the last character in the row).

Playfair Cipher

b. If the two letters in a pair are located in the same column of the secret key, the cor responding encrypted character for each letter is the letter beneath it in the same column (with wrapping to the beginning of the column if the plaintext letter is the last character in the column).

c. If the two letters in a pair are not in the same row or column of the secret, the cor responding encrypted character for each letter is a letter that is in its own row but in the same column as the other letter.

Playfair Cipher

$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3 \ldots$ $k = [(k_1, k_2), (k_3, k_4), \ldots]$

Encryption: $C_i = k_i$ Decryption: $P_i = k_i$

Example

Let us encrypt the plaintext "hello" using the key in Figure 3.13. When we group the letters in two-character pairs, we get "he, ll, o". We need to insert an x between the two l's (els), giving "he, lx, lo". We have

he → EC lx → QZ lo → BX

Plaintext: hello Ciphertext: ECQZBX

Vigenere Cipher

- A Vigenere cipher uses a different strategy to create the key stream
- The key stream is a repetition of an initial secret key stream of length m, where we have 1 ≤ m ≤ 26
- The cipher can be described as follows where (k 1 k 2 , …, k m) is the initial secret key agreed to by Alice and Bob.

$$P = P_1P_2P_3 \ldots \qquad C = C_1C_2C_3 \ldots \qquad K = [(k_1, k_2, \ldots, k_m), (k_1, k_2, \ldots, k_m), \ldots]$$

$$\text{Encryption: } C_i = P_i + k_i \qquad \text{Decryption: } P_i = C_i - k_i$$

- One important difference between the Vigenere cipher and the other two poly alphabetic ciphers, is that the Vigenere key stream does not depend on the plaintext characters.
- It depends only on the position of the character in the plaintext
- In other words, the key stream can be created without knowing what the plaintext is.
- Example
  - Encrypt the message “She is listening” using the 6-character keyword “PASCAL”. The initial key stream is (15, 0, 18, 2, 0, 11). The key stream is the repetition of this initial key stream (as many times as needed).

Plaintext:	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P's values:	18	07	04	08	18	11	08	18	19	04	13	08	13	06
Key stream:	***15***	***00***	***18***	***02***	***00***	***11***	***15***	***00***	***18***	***02***	***00***	***11***	***15***	***00***
C's values:	**07**	**07**	**22**	**10**	**18**	**22**	**23**	**18**	**11**	**6**	**13**	**19**	**02**	**06**
Ciphertext:	**H**	**H**	**W**	**K**	**S**	**W**	**X**	**S**	**L**	**G**	**N**	**T**	**C**	**G**

Hill Cipher

- the Hill cipher belongs to a category of ciphers called block ciphers.
- The other ciphers we studied so far belong to the category called stream ciphers.
- Here the plaintext is divided into equal-size blocks.
- The blocks are encrypted one at a time in such a way that each character in the block contributes to the encryption of other char acters in the block

- In a Hill cipher, the key is a square matrix of size m × m in which m is the size of the block.
- If we call the key matrix K, each element of the matrix is k_{ij} as shown

Key in the Hill cipher

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \dots & k_{mm} \end{bmatrix}$$

- Let us see how one block of the ciphertext is encrypted
- If we call the m characters in the plaintext block P1,P2 P3…,Pm . the corresponding characters in the ciphertext block are C1 ,C2, …,Cm Then we have

$$
\begin{aligned}
C_1 &= P_1 k_{11} + P_2 k_{21} + \cdots + P_m k_{m1} \\
C_2 &= P_1 k_{12} + P_2 k_{22} + \cdots + P_m k_{m2} \\
&\cdots \\
C_m &= P_1 k_{1m} + P_2 k_{2m} + \cdots + P_m k_{mm}
\end{aligned}
$$

- Bob will not be able to decrypt the ciphertext sent by Alice if the matrix does not have a multiplicative inverse.

- Example :
  - the plaintext “code is ready” can make a 3× 4 matrix when adding extra bogus character “z” to the last block and removing the spaces.
  - And The ciphertext is “OHKNIHGKLISS”. Bob can decrypt the message using the inverse of the key matrix

$$\begin{matrix}\mathbf{C}\\ \begin{bmatrix}14 & 07 & 10 & 13\\ 08 & 07 & 06 & 11\\ 11 & 08 & 18 & 18\end{bmatrix}\end{matrix} = \begin{matrix}\mathbf{P}\\ \begin{bmatrix}02 & 14 & 03 & 04\\ 08 & 18 & 17 & 04\\ 00 & 03 & 24 & 25\end{bmatrix}\end{matrix} \begin{matrix}\mathbf{K}\\ \begin{bmatrix}09 & 07 & 11 & 13\\ 04 & 07 & 05 & 06\\ 02 & 21 & 14 & 09\\ 03 & 23 & 21 & 08\end{bmatrix}\end{matrix}$$

a. Encryption

$$\begin{matrix}\mathbf{P}\\ \begin{bmatrix}02 & 14 & 03 & 04\\ 08 & 18 & 17 & 04\\ 00 & 03 & 24 & 25\end{bmatrix}\end{matrix} = \begin{matrix}\mathbf{C}\\ \begin{bmatrix}14 & 07 & 10 & 13\\ 08 & 07 & 06 & 11\\ 11 & 08 & 18 & 18\end{bmatrix}\end{matrix} \begin{matrix}\mathbf{K}^{-1}\\ \begin{bmatrix}02 & 15 & 22 & 03\\ 15 & 00 & 19 & 03\\ 09 & 09 & 03 & 11\\ 17 & 00 & 04 & 07\end{bmatrix}\end{matrix}$$

b. Decryption

One-Time Pad

- One of the goals of cryptography is perfect secrecy.
- A study by Shannon has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain . For example
  - Example of per-character keys:
  - 1st character ⟶ key **04**
  - 2nd character ⟶ key **02**
  - 3rd character ⟶ key **21**
  - … and so on

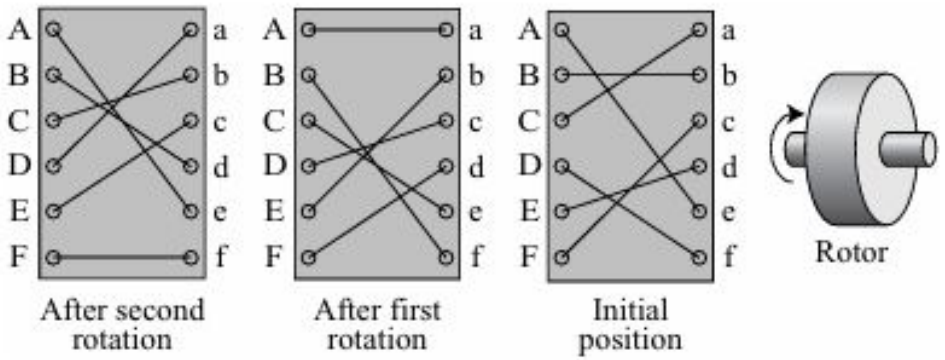
With random, per-character keys, the ciphertext reveals **no statistical information** about the plaintext. As a result, **ciphertext-only attacks become impossible**.

- This idea is used in a cipher called ***one-time pad***, invented by Vernam. The key has the same length as the plaintext and is chosen completely in random.
- A one-time pad is a perfect cipher, but it is almost impossible to implement commercially.
- However, there are some occasions when a one-time pad can be used
  - For example, if the president of a country needs to send a completely secret message to the president of another country, she can send a trusted envoy with the random key before sending the message.

Rotor Cipher

- Although one-time pad ciphers are not practical, one step toward more secured encipherment is the ***rotor cipher.***
- It uses the idea behind monoalphabetic substitution but changes the mapping between the plaintext and the ciphertext characters for each plain text character.

A rotor cipher



- The initial setting (position) of the rotor is the secret key between Alice and Bob.
- The first plaintext character is encrypted using the initial setting; the second character is encrypted after the first rotation and so on.
- A three-letter word such as “bee” is encrypted as “BAA” if the rotor is stationary (the monoalphabetic substitution cipher), but it will encrypted as “BCA” if it is rotating (the rotor cipher).
- This shows that the rotor cipher is a polyalphabetic cipher because two occurrences of the same plaintext character are encrypted as different characters.

Transposition Cipher

- A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols
- A symbol in the first position of the plaintext may appear in the tenth position of the ciphertext, and A symbol in the eighth position in the plaintext may appear in the first position of the ciphertext.
- In other words, a transposition cipher reorders (transposes) the symbols.

Keyless Transposition Ciphers

- Simple transposition ciphers, which were used in the past, are keyless.
- There are two methods for permutation of characters
- In the first method, the text is written into a table column by column and then transmitted row by row.
- In the second method, the text is written into the table row by row and then transmitted column by column.

Example

- A good example of a keyless cipher using the first method is the ***rail fence cipher***
- In this cipher, the plaintext is arranged in two lines as a zigzag pattern (which means column by column);
- the ciphertext is created reading the pattern row by row
- For example, to send the message “Meet me at the park” to Bob, Alice writes

- She then creates the ciphertext “ MEMATEAKETETHPR” by sending the first row followed by the second row

- Bob receives the ciphertext and divides it in half (in this case the sec ond half has one less character).
- The first half forms the first row; the second half, the second row. Bob reads the result in zigzag.
- Because there is no key and the number of rows is fixed (2), the cryptanalysis of the ciphertext would be very easy for Eve.
- All she needs to know is that the ***rail fence cipher*** is used.

- Second method: Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.

m	e	e	t
m	e	a	t
t	h	e	p
a	r	k	

  - She then creates the ciphertext " MMTAEEHREAEKTTP" by transmitting the characters column by column.

- Bob receives the ciphertext and follows the reverse process. He writes the received message, column by column, and reads it row by row as the plaintext. Eve can easily decipher the message if she knows the number of columns.

Keyed Transposition Ciphers

- The keyless ciphers permute the characters by using writing plaintext in one way (row by row, for example) and reading it in another way (column by column, for example).
- 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

- Alice needs to send the message “Enemy attacks tonight” to Bob.
- Alice and Bob have agreed to divide the text into groups of five characters and then permute the characters in each group.
- The following shows the grouping after adding a bogus character at the end to make the last group the same size as the others

e n e m y a t t a c k s t o n i g h t z

- The key used for encryption and decryption is a permutation key, which shows how the character are permuted.
- For this message, assume that Alice and Bob used the following key:

Encryption ↓						↑ Decryption
	3	1	4	5	2	
	1	2	3	4	5	

- The third character in the plaintext block becomes the first character in the ciphertext block; the first character in the plaintext block becomes the second character in the ciphertext block; and so on. The permutation yields

E E M Y N T A A C T T K O N S H I T Z G

STREAM AND BLOCK CIPHERS

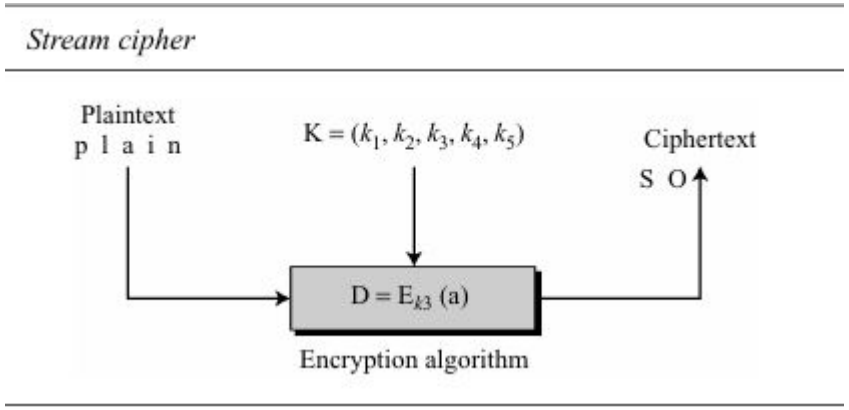
Stream Ciphers

- In a stream cipher, encryption and decryption are done one symbol (such as a charac ter or a bit) at a time.
- We have a plaintext stream, a ciphertext stream, and a key stream.
- Call the plaintext stream P, the ciphertext stream C, and the key stream K.

$$P = P_1P_2P_3, \ldots \qquad C = C_1C_2C_3, \ldots \qquad K = (k_1, k_2, k_3, \ldots)$$

$$C_1 = E_{k1}(P_1) \qquad C_2 = E_{k2}(P_2) \qquad C_3 = E_{k3}(P_3) \ldots$$

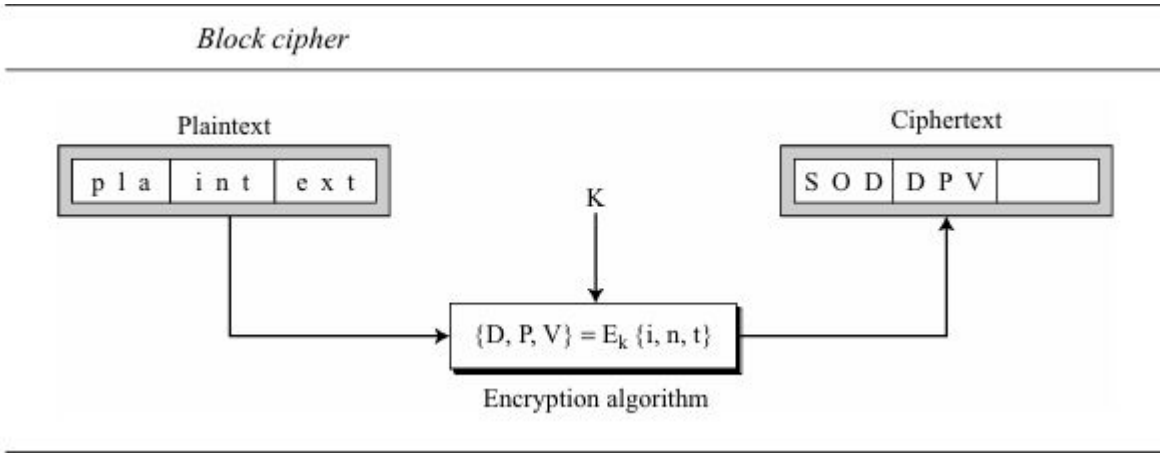
- Characters in the plaintext are fed into the encryption algorithm, one at a time; the ciphertext characters are also created one at a time.
- The key stream, can be created in many ways. It may be a stream of predetermined values;
- It may be created one value at a time using an algorithm.
- The values may depend on the plaintext or ciphertext characters. The values may also depend on the previous key values.



Block Ciphers

In a block cipher, a group of plaintext symbols of size m (m > 1) are encrypted together creating a group of ciphertext of the same size.

Based on the definition, in a block cipher, a single key is used to encrypt the whole block even if the key is made of multiple values.



In a block cipher, a ciphertext block depends on the whole plaintext block.

THANK YOU

Archana M

Department of Computer Applications

archanam@pes.edu