



1495  
UNIVERSITY OF  
ABERDEEN

CELEBRATING  
525 YEARS  
1495 – 2020

ABERDEEN 2040

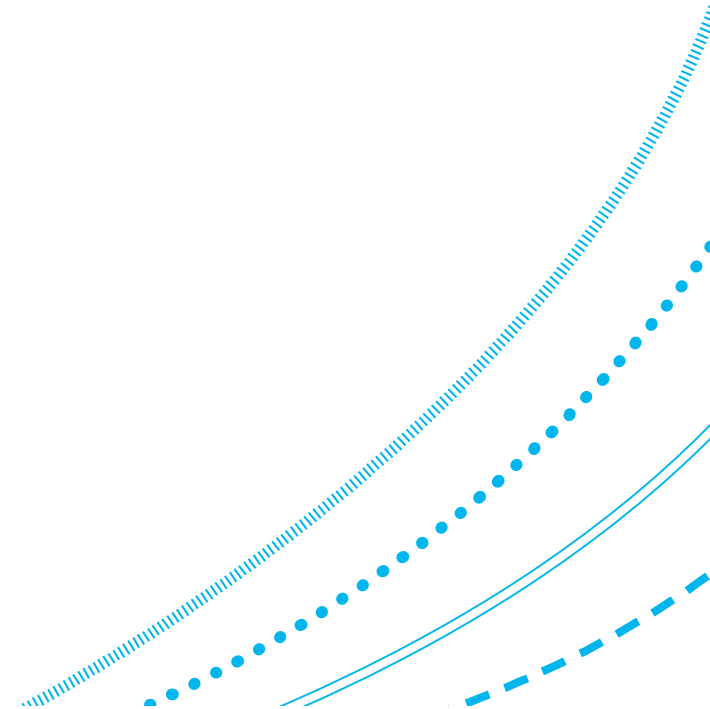
# Network Security Technology

## Introduction to Cryptography

September 2025

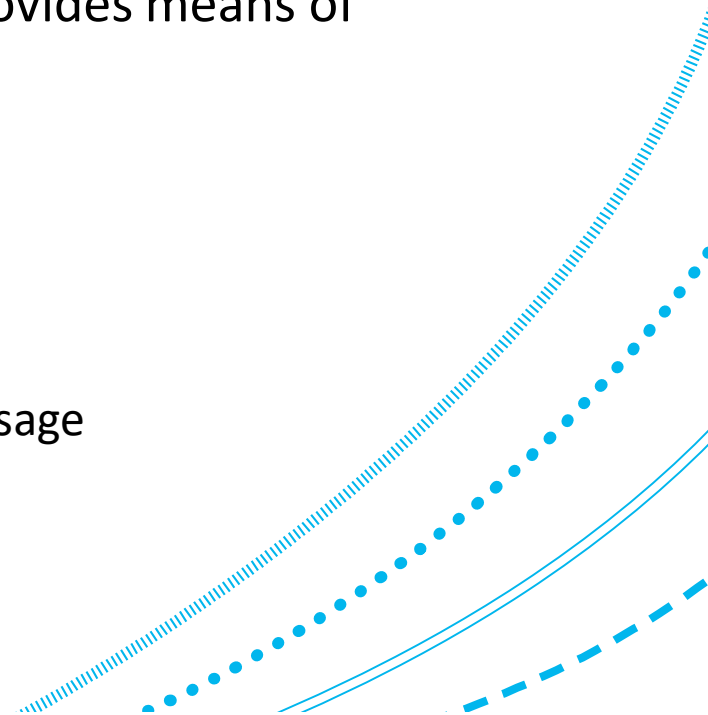
# Outline of lecture

1. Four goals
2. Terminologies
3. Technique classification
4. Classical (Traditional) ciphers
  1. Substitution cipher
  2. Transposition cipher
  3. Product cipher
5. Block Ciphers and Stream Ciphers
6. Cryptographic modes



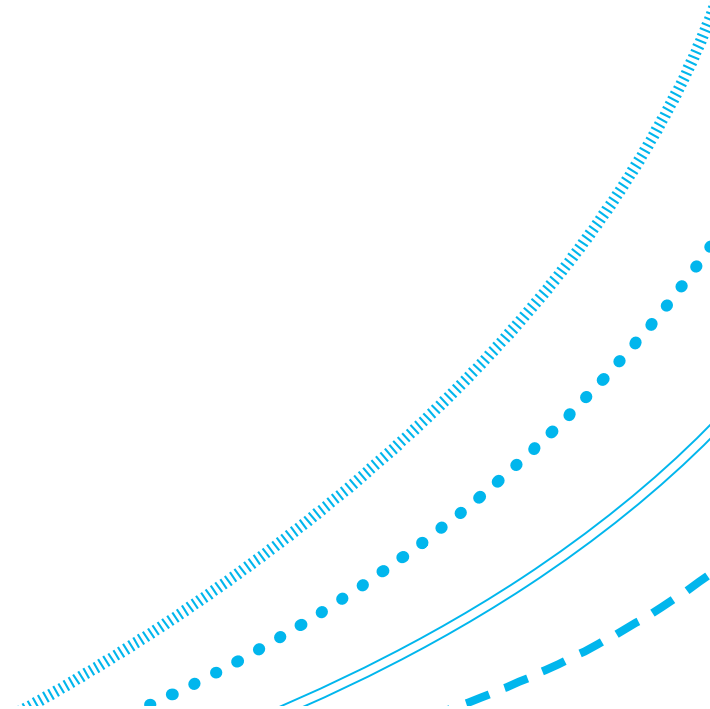
# Definitions

1. crypto - hidden/secret + grafia - writing
2. “the science and study of secret writing”
3. Cryptography is the science of protecting data, which provides means of converting data into unreadable form, so that
  1. the data *cannot be accessed* for *unauthorised use*
  2. the content of the data frames *is hidden*
  3. the authenticity of the data *can be established*
  4. the undetected modification of the data *is avoided*
  5. the data *cannot be disowned* by the originator of the message



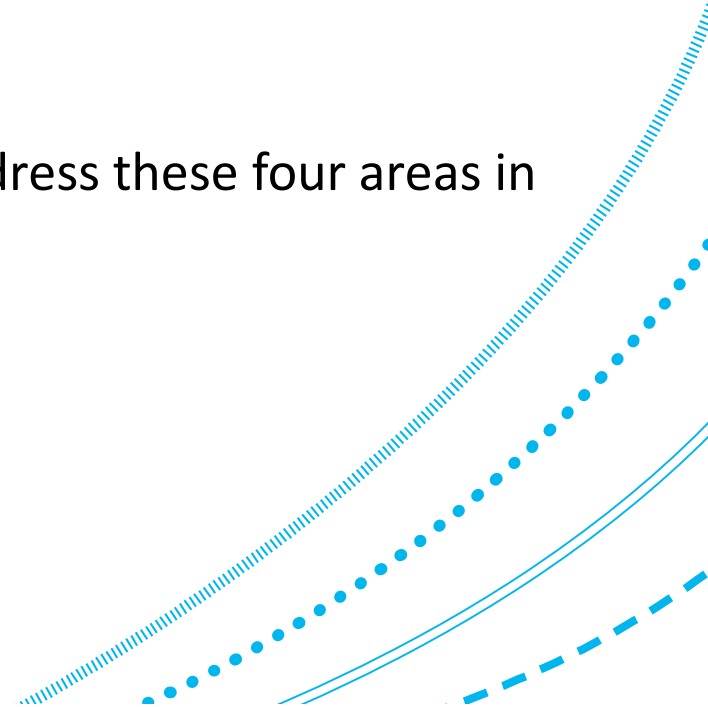
# Why cryptography?

1. Passing (secret) information through potentially **insecure** channels, such as:
  1. military campaign,
  2. financial/online banking/shopping,
  3. diplomatic communications (espionage).



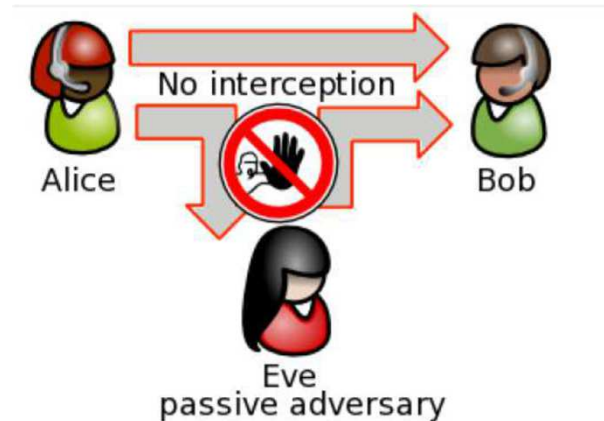
# Four goals of cryptography

1. In spite of adversaries, we want to achieve (among other things):
  1. **Confidentiality** - prevent unauthorised access;
  2. **Integrity** - no modification of existing information;
  3. **Authentication** - no identifying either entities or data origins;
  4. **Non-repudiation** - preventing denials of messages sent
2. A fundamental goal of cryptography is to adequately address these four areas in both theory and practice.



# Goal 1: Confidentiality

1. This comprises two separate requirements:
  1. no observer can access the contents of the message.
  2. no observer can identify the sender and receiver.
2. The terms **privacy** or **secrecy** are also used to mean confidentiality



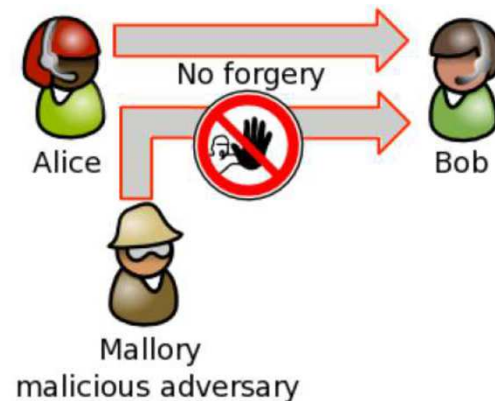
## Goal 2: Integrity

1. This requires that the recipient can be sure that:
  1. the message has not been changed or lost during transmission.
  2. the message has not been prevented from reaching the recipient.
  3. the message has not reached the recipient twice.



# Goal 3: Authentication

1. This requires that:
  1. the sender can be sure that the message reaches the intended recipient, and only the intended recipient, and
  2. the recipient can be sure that the message came from the sender and not an imposter. The act by an imposter of sending such a message is referred to as “spoofing”





## Goal 4: Non-repudiation

1. This requires that:
  1. the sender cannot deny that the message was sent by him.
  2. the recipient cannot deny that the message was received by him.

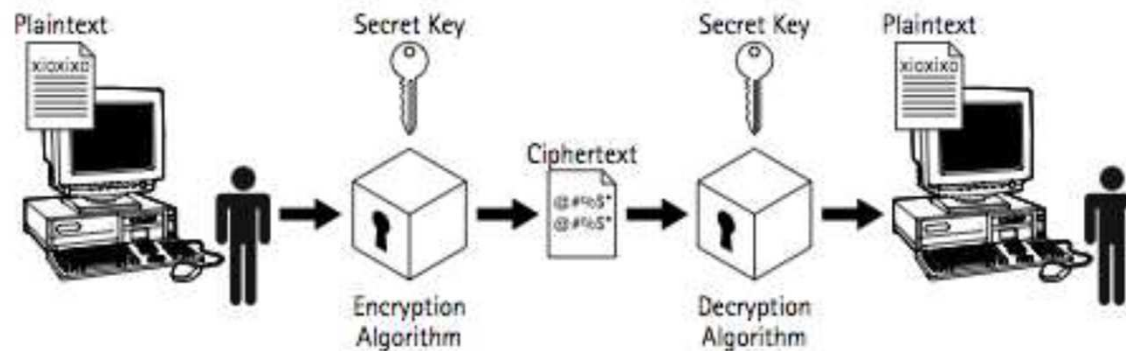


# Terminology

1. **Plain text (or clear text)** - text that can be read by a human
2. **Encryption** - process of transforming plaintext into ciphertext
3. **Cipher text (or encrypted text)** - text that needs to be processed to be read by a human being
4. **Decryption** - process of transforming a cipher text into a plain text (the reverse of encryption)
5. **Cipher** - a secret method of writing (i.e., encryption scheme: mathematical function(s) or algorithm(s) used for encryption and decryption, they are usually using **keys**)
6. **Key** - is a word, number, or phrase that is used to encrypt the clear text.

# Conventional encryption model

1. A sender wants to send a “hello” message to a recipient:
  1. The original message (plaintext) is converted to ciphertext by using a key and an algorithm.
  2. The ciphertext is transmitted over the transmission medium.
  3. At the recipient end, the ciphertext is converted back to the original text using the same algorithm and key that were used to encrypt the message.



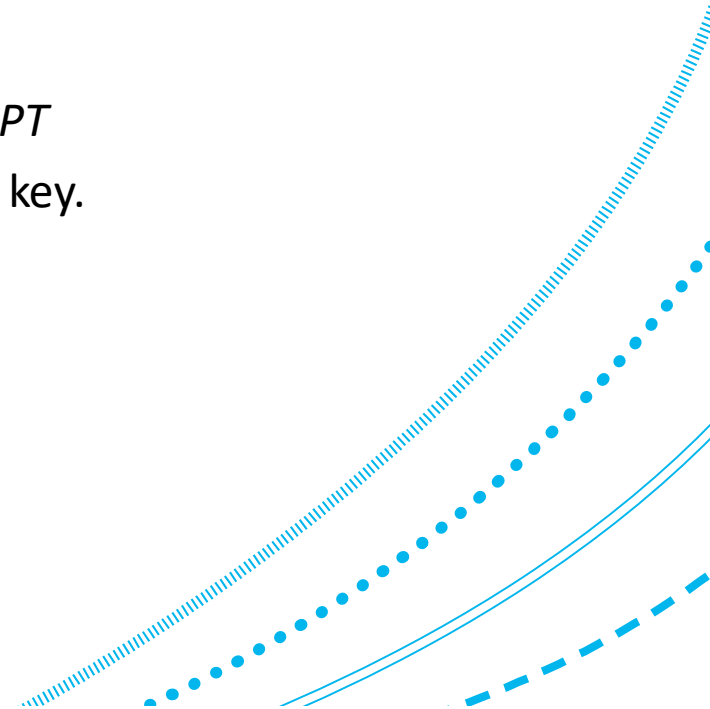
# Main cryptography techniques

## 1. Symmetric encryption:

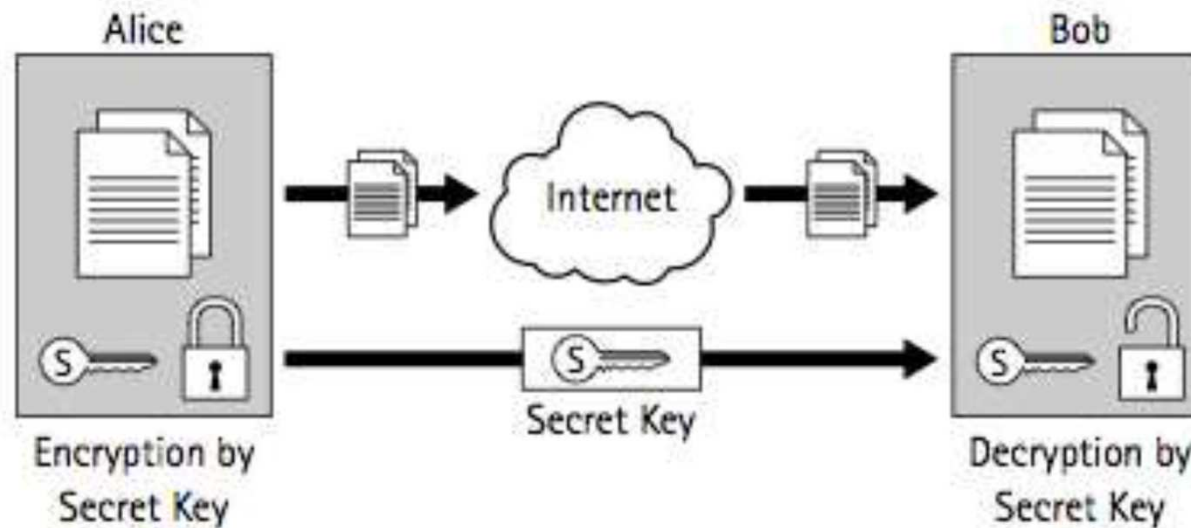
1. known as **secret** key cryptography:  $E_k(PT) = CT, D_k(CT) = PT$
2. based on a **single key**: the same key is used to encrypt and decrypt the data

## 2. Non-symmetric encryption:

1. known as **public** key cryptography:  $E_{k_1}(PT) = CT, D_{k_2}(CT) = PT$
2. base on a combination of **two keys** - secret key and public key.
3. public key is used for encryption, and
4. the secret key is used for decryption.

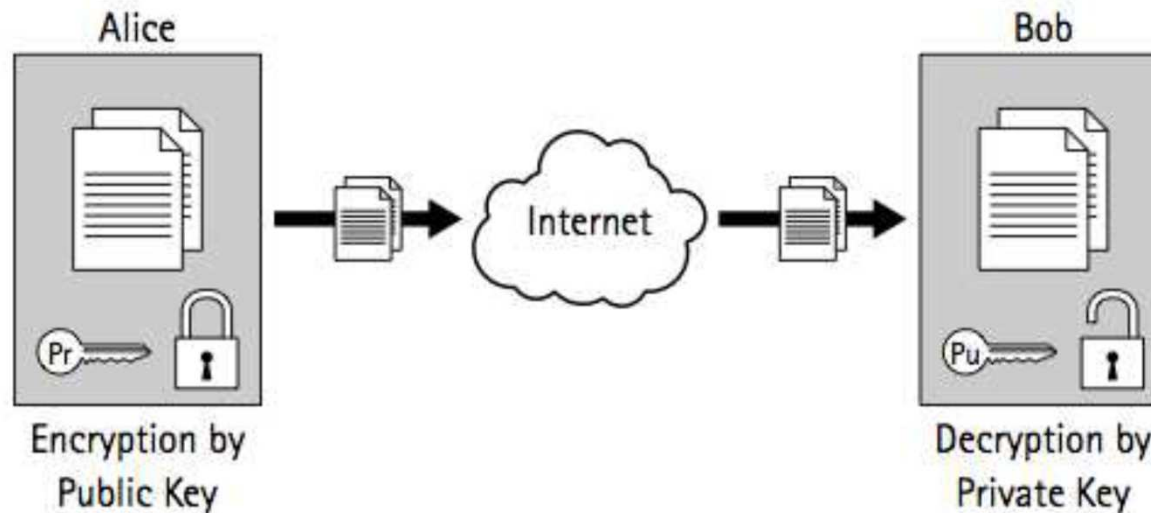


# Symmetric encryption



- Main problem: the sender and the receiver have to agree on a common key, a secure channel is also required exchange the secret key.
- Most widely used secret key algorithms are DES, 3DES, AES.

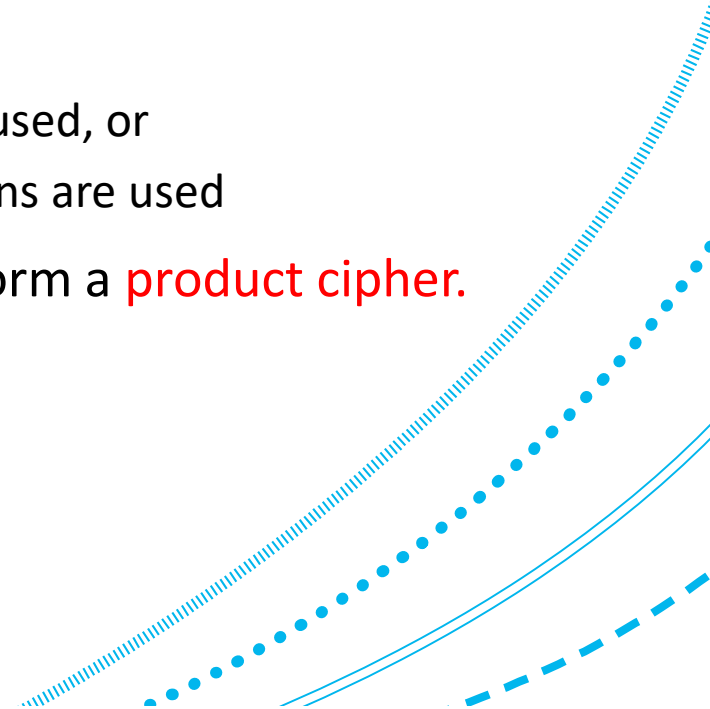
# Non-symmetric encryption



1. Non-symmetric: both keys are required to complete the process (encrypted by the public key and decrypted by the private key).
2. Widely used algorithm: RSA.

# Traditional cryptographic techniques

1. Two basic components of classical ciphers: **substitution** and **transposition**
  1. Substitution ciphers: letters are replaced by other letters
  2. Transposition ciphers: the letters are arranged in a different order
2. These ciphers may be:
  1. **Monoalphabetic** - only one substitution/ transposition is used, or
  2. **Polyalphabetic** - where several substitutions/ transpositions are used
3. Several such ciphers may be concatenated together to form a **product cipher**.



# Substitution cipher

## 1. Caesar (50-60BC) – monoalphabetic:

1. ignore space character, gather letters in t-letter blocks
2. rotate left or right by some number of positions to obtain cipher text.
3. can describe this cipher as:
  1. Encryption  $E_k : i \rightarrow i + k \bmod 26$
  2. Decryption  $D_k : i \rightarrow i - k \bmod 26$

plain text	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
cipher text	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

Example:

plain text	RETURN TO ROME	cipher text	VIXYV RXSVS QI
5-letter blocks	RETUR NTORO ME	-4 shift	RETUR NTORO ME
+4 shift	VIXYV RXSVS QI	interpretation	RETURN TO ROME



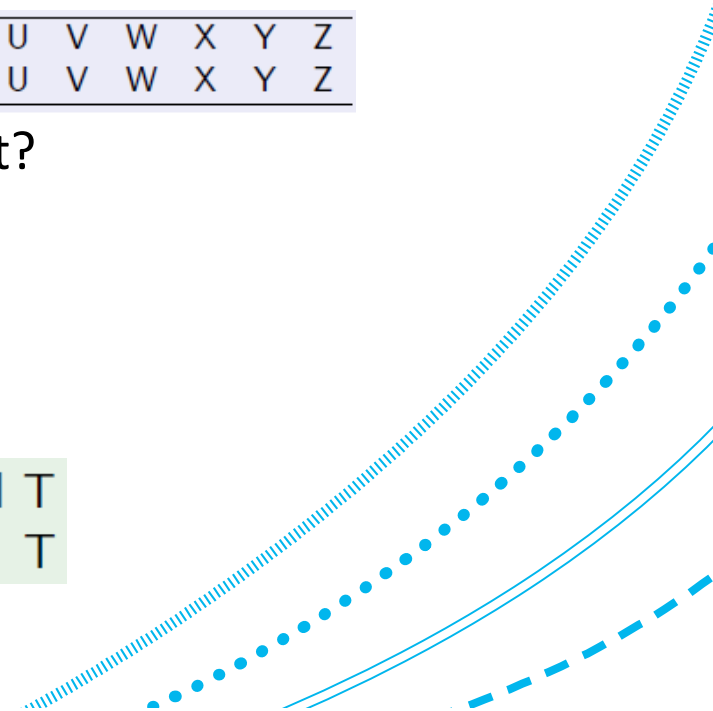
# Substitution cipher (cont'd)

- Caesar cipher generalisation, keyword used to permute the alphabet:
- Write keyword (no repeat characters), suppose keyword is JACKSON, followed by remainder of alphabet in order:

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

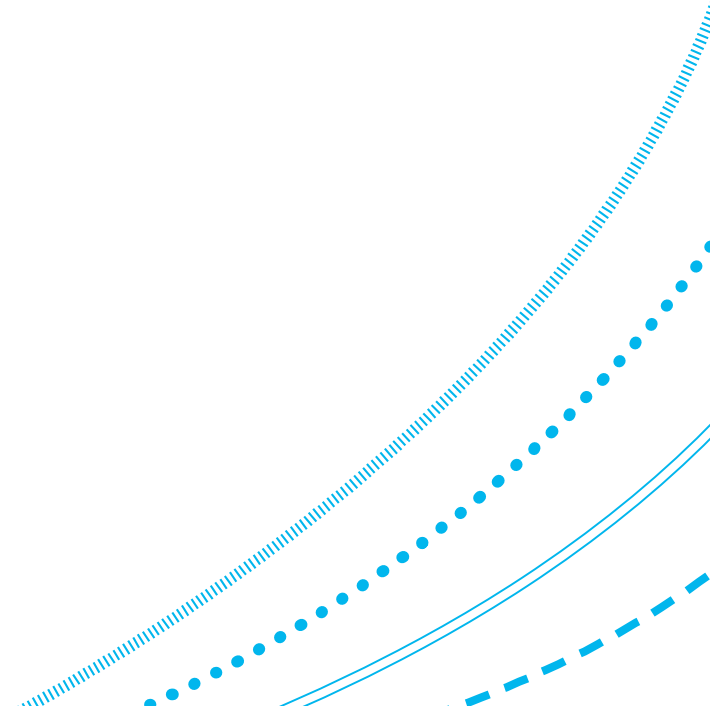
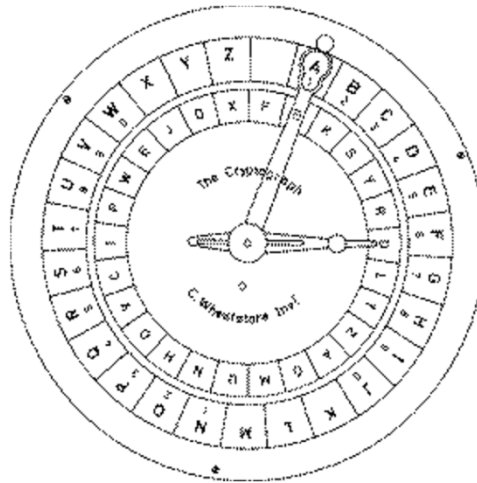
- What does your partner in crime need to encrypt/decrypt?
  - just the keyword.
- How secure? i.e., how difficult to break?
  - letter frequency analysis is a good attack.

plain text	S	E	N	D	R	E	I	N	F	O	R	C	E	M	E	N	T
cipher text	R	S	I	K	Q	S	D	I	O	L	Q	C	S	H	S	I	T



# Substitution cipher (cont'd)

1. Porta (1563) – polyalphabetic (polygraphic):
  1. Replace 2-letter blocks with corresponding symbols
  2. The first letter (key) is stationary while the second letter moves, indicating which symbol is to be used instead of the original 2-letter block.



# Substitution cipher (cont'd)

1. Porta: A matrix can easily represent the original disc ...

Keys	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,B		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
C,D		o	p	q	r	s	t	u	v	w	x	y	z	n	m	a	b	c	d	e	f	g	h	i	j	k	l
E,F		p	q	r	s	t	u	v	w	x	y	z	n	o	l	m	a	b	c	d	e	f	g	h	i	j	k
G,H		q	r	s	t	u	v	w	x	y	z	n	o	p	k	l	m	a	b	c	d	e	f	g	h	i	j
I,J		r	s	t	u	v	w	x	y	z	n	o	p	q	j	k	l	m	a	b	c	d	e	f	g	h	i
K,L		s	t	u	v	w	x	y	z	n	o	p	q	r	i	j	k	l	m	a	b	c	d	e	f	g	h
M,N		t	u	v	w	x	y	z	n	o	p	q	r	s	h	i	j	k	l	m	a	b	c	d	e	f	g
O,P		u	v	w	x	y	z	n	o	p	q	r	s	t	g	h	i	j	k	l	m	a	b	c	d	e	f
Q,R		v	w	x	y	z	n	o	p	q	r	s	t	u	f	g	h	i	j	k	l	m	a	b	c	d	e
S,T		w	x	y	z	n	o	p	q	r	s	t	u	v	e	f	g	h	i	j	k	l	m	a	b	c	d
U,V		x	y	z	n	o	p	q	r	s	t	u	v	w	d	e	f	g	h	i	j	k	l	m	a	b	c
W,X		y	z	n	o	p	q	r	s	t	u	v	w	x	c	d	e	f	g	h	i	j	k	l	m	a	b
Y,Z		z	n	o	p	q	r	s	t	u	v	w	x	y	b	c	d	e	f	g	h	i	j	k	l	m	a

2. The 'key' for a porta cipher is a key word. e.g. 'FORTIFICATION'
3. To encipher a message, repeat the keyword above the plaintext.

# Substitution cipher (cont'd)

Keys	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,B	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
C,D	o	p	q	r	s	t	u	v	w	x	y	z	n	m	a	b	c	d	e	f	g	h	i	j	k	l
E,F	p	q	r	s	t	u	v	w	x	y	z	n	o	l	m	a	b	c	d	e	f	g	h	i	j	k
G,H	q	r	s	t	u	v	w	x	y	z	n	o	p	k	l	m	a	b	c	d	e	f	g	h	i	j
I,J	r	s	t	u	v	w	x	y	z	n	o	p	q	j	k	l	m	a	b	c	d	e	f	g	h	i
K,L	s	t	u	v	w	x	y	z	n	o	p	q	r	i	j	k	l	m	a	b	c	d	e	f	g	h
M,N	t	u	v	w	x	y	z	n	o	p	q	r	s	h	i	j	k	l	m	a	b	c	d	e	f	g
O,P	u	v	w	x	y	z	n	o	p	q	r	s	t	g	h	i	j	k	l	m	a	b	c	d	e	f
Q,R	v	w	x	y	z	n	o	p	q	r	s	t	u	f	g	h	i	j	k	l	m	a	b	c	d	e
S,T	w	x	y	z	n	o	p	q	r	s	t	u	v	e	f	g	h	i	j	k	l	m	a	b	c	d
U,V	x	y	z	n	o	p	q	r	s	t	u	v	w	d	e	f	g	h	i	j	k	l	m	a	b	c
W,X	y	z	n	o	p	q	r	s	t	u	v	w	x	c	d	e	f	g	h	i	j	k	l	m	a	b
Y,Z	z	n	o	p	q	r	s	t	u	v	w	x	y	b	c	d	e	f	g	h	i	j	k	l	m	a

Porta cipher (cont'd):  
Example:

key	F	O	R	T	I	F	I	C	A	T	I	O	N	F	O	R	T	I	F	I	C	A	T	I	O	N	F	O
plain text	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
cipher text	s	y	n	n	j	s	c	v	r	n	r	l	a	h	u	t	u	k	u	c	v	r	y	r	l	a	n	y

# Substitution cipher (cont'd)

1. Vigenere (1553) – polyalphabetic:
  1. The message is encrypted using the original plain text, a (text) key, and the table.

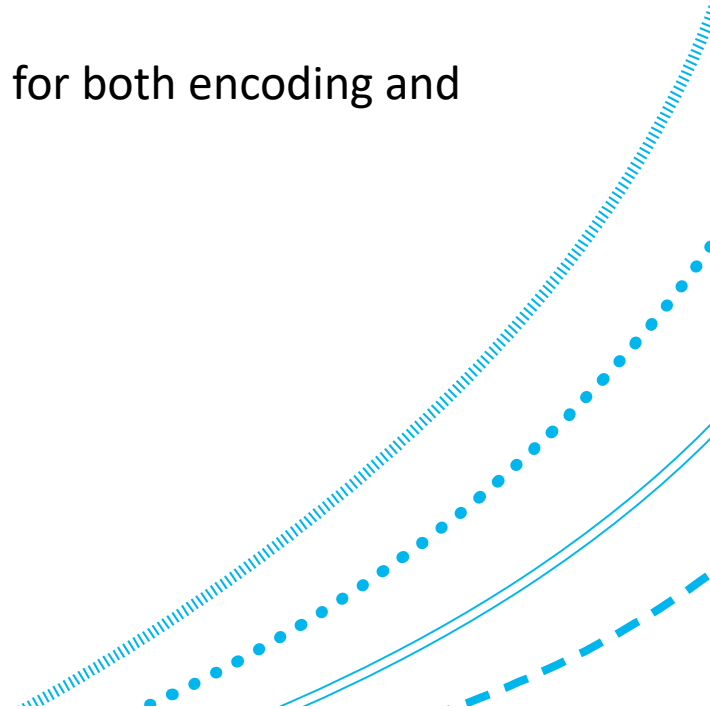
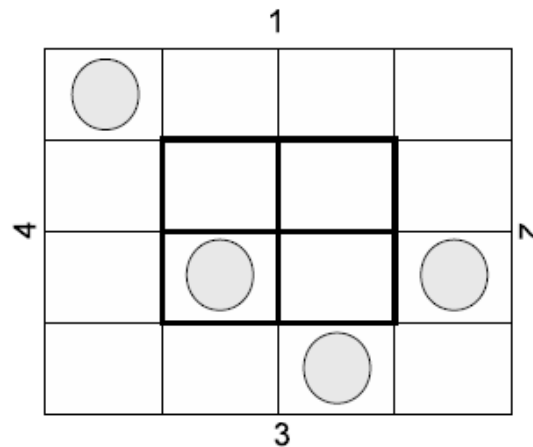
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T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

Example:

plain text	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
key	C	R	Y	P	T	O	C	R	Y	P	T	O	C	R	Y	P	T	O	C	R	Y	P	T	O	C	R
cipher text	C	S	A	S	X	T	I	Y	G	Y	D	Z	O	E	M	E	J	F	U	K	S	K	P	L	A	Q

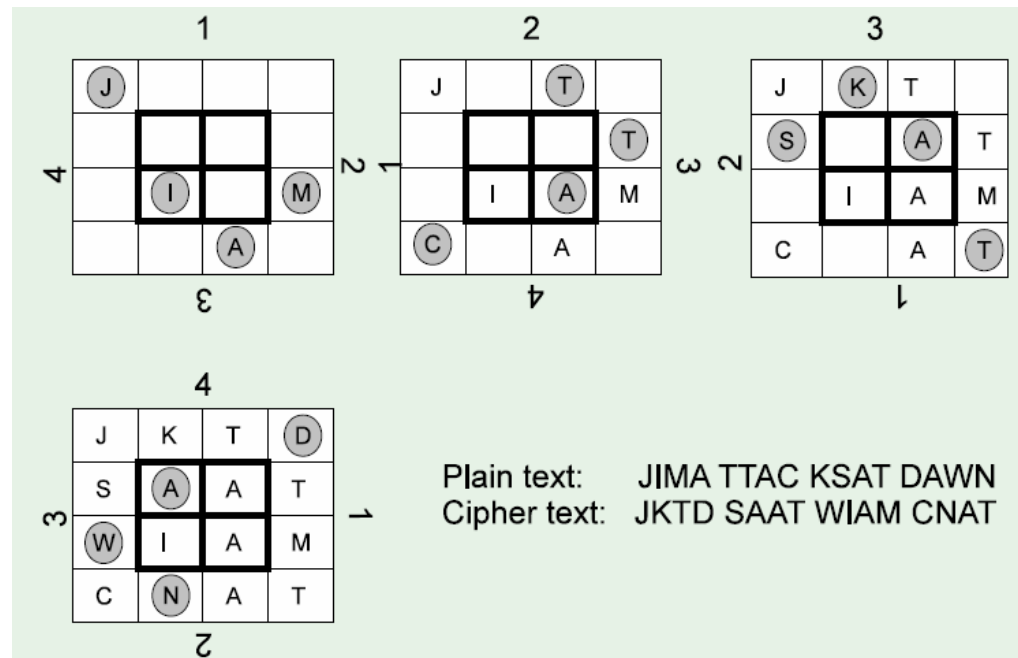
# Transposition cipher

1. Turning Grille (Fleissner, Wostrowitz 1881):
  1. This template was a square with a number of holes punched out.
  2. There are an even number of rows and columns (thus the total number of fields is divisible by 4).
  3. One fourth of these fields is cut out. This template is used for both encoding and decoding the message.



# Transposition cipher (cont'd)

1. Encrypt JIM ATTACKS AT DAWN using this grille.



2. This system was used in World War II by German spies in South America.

# Product cipher

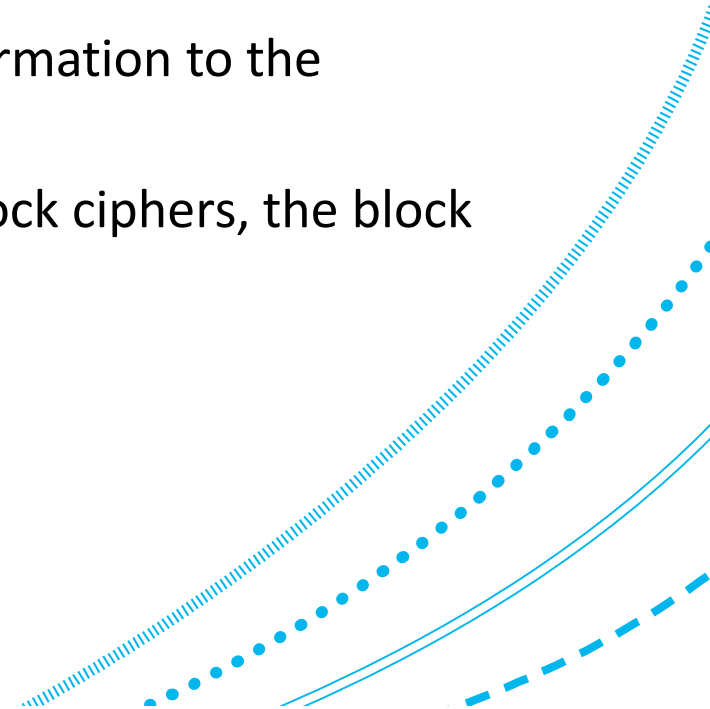
1. Feistel-IBM-1971:
  1. Predecessor for the Data Encryption Standard (DES).
  2. This system uses permutations (transpositions) on large blocks for the mixing transformation, and substitution on small blocks for confusion.
  3. This system is based on two hardware components:
    1. P-box (Permutation box)
    2. S-box (Substitution box)

We will study the Feistel cipher and DES, which is based on Feistel, in the Symmetric encryption lecture.



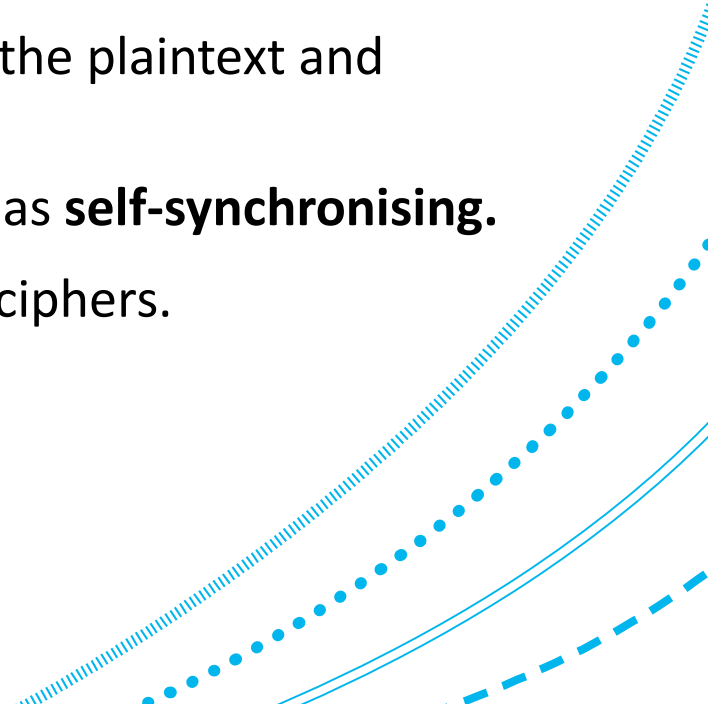
# Block ciphers

1. A type of symmetric-key encryption.
2. Transforms a fixed-length block of plaintext into a block of ciphertext of the same length, using a user provided secret key.
3. Decryption is performed by applying the reverse transformation to the ciphertext block using the same secret key.
4. The fixed length is called the block size, and for many block ciphers, the block size is 64 bits.



# Stream ciphers

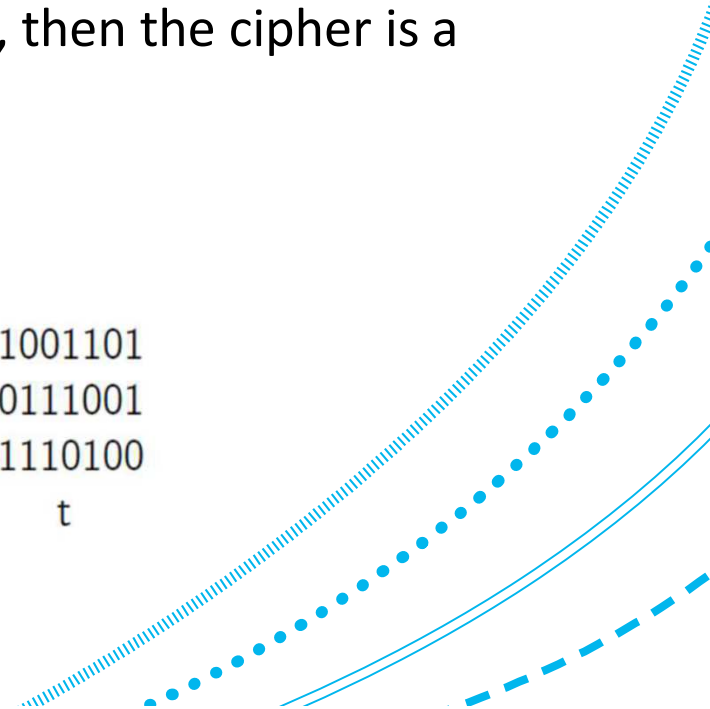
1. A stream cipher generates a **keystream**, a sequence of bits used as a key.
2. Encryption: accomplished by combining the **keystream** with the plaintext, usually with the bitwise **XOR** operation.
3. The generation of the keystream can be independent of the plaintext and ciphertext, termed as **synchronous**.
4. Or it can depend on the data and its encryption, termed as **self-synchronising**.
5. Most stream cipher designs are for synchronous stream ciphers.



# Stream ciphers (cont'd)

1. A **Vernam cipher** is a stream cipher in which the plaintext is *XORed* with a random or pseudorandom stream of data of the same length to generate the ciphertext.
2. If the stream of data is truly random and used only once, then the cipher is a one-time pad.

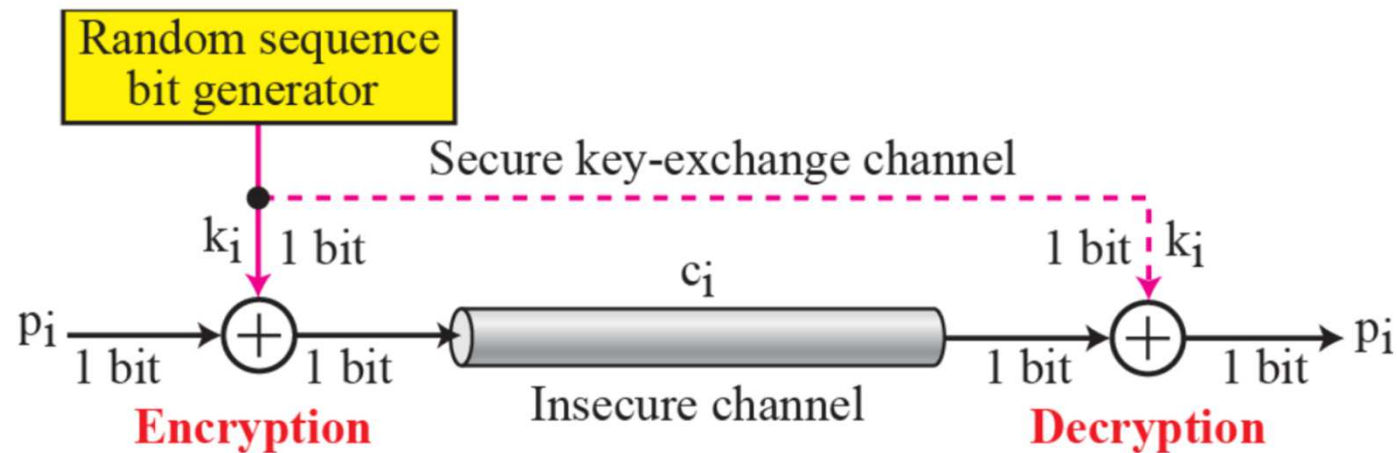
Encryption			Decryption		
c	a	t	c	a	t
01100011	01100001	01110100	11010010	00110100	11001101
$\oplus$ 10110001	01010101	10111001	$\oplus$ 10110001	01010101	10111001
11010010	00110100	11001101	01100011	01100001	01110100



# Stream ciphers (cont'd)

## 1. One Time Pad:

1. The message is encrypted by combining (usually XORing) it with a perfectly random key at least as long as the message and the key is only used once.
2. Apart from the problem of obtaining a perfectly random key, the main problem with one-time pads is the distribution of keys.



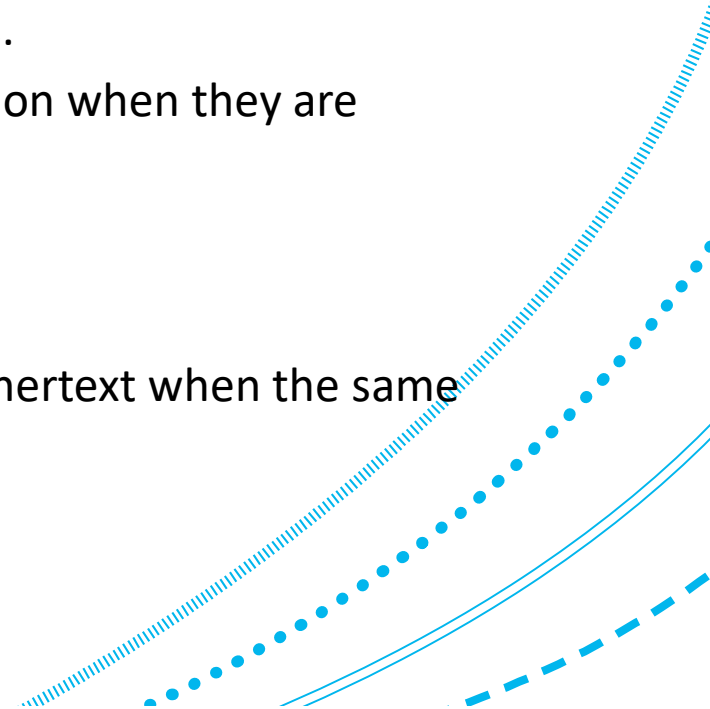
# Stream ciphers vs. block ciphers

## 1. Stream cipher:

1. A type of symmetric encryption algorithm.
2. Can be designed to be exceptionally fast, much faster than any block cipher.
3. Typically operate on smaller units of plaintext, usually bits.
4. The transformation of plaintext units will vary, depending on when they are encountered during the encryption process.

## 2. Block cipher:

1. Operate on large blocks of data.
2. The encryption of any plaintext will result in the same ciphertext when the same key is used.

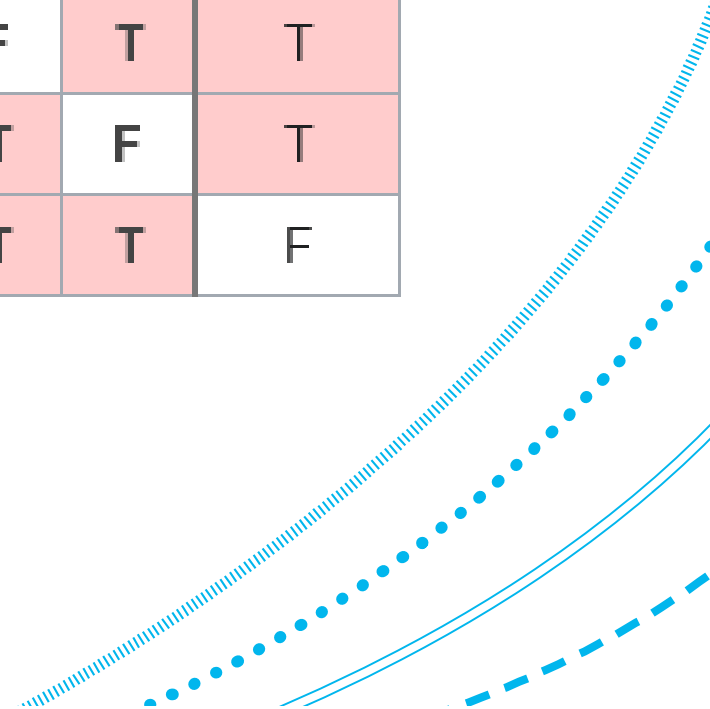


# XOR

1. XOR stands for “Exclusive OR”, denoted by  $\oplus$ .
2. With two inputs, XOR is TRUE **if and only if** the inputs differ (one is TRUE, one is FALSE).
3. With multiple inputs, XOR is TRUE **if and only if** the number of true inputs is odd.

$A \nabla$	$B \nabla$	$A \oplus B$
F	F	F
F	T	T
T	F	T
T	T	F

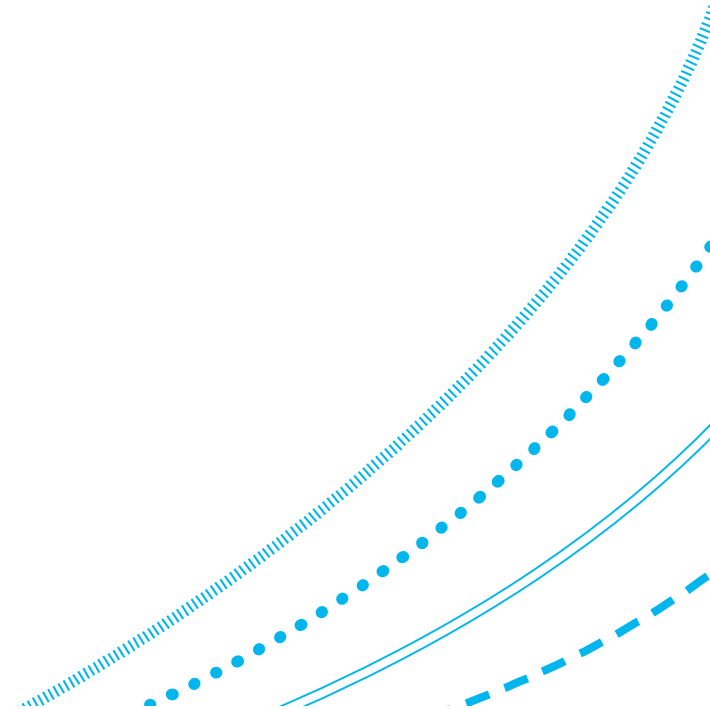
Reference: [https://en.wikipedia.org/wiki/Exclusive\\_or](https://en.wikipedia.org/wiki/Exclusive_or)



# XOR Example 1

1.  $4 \oplus 3$
2. Step 1: convert 4 into binary form, 0100
3. Step 2: convert 3 into binary form, 0011
4. Step 3: the XOR operation is to XOR each bit, so

$$\begin{array}{r} 0100 \\ \oplus \underline{0011} \\ 0111 \end{array}$$



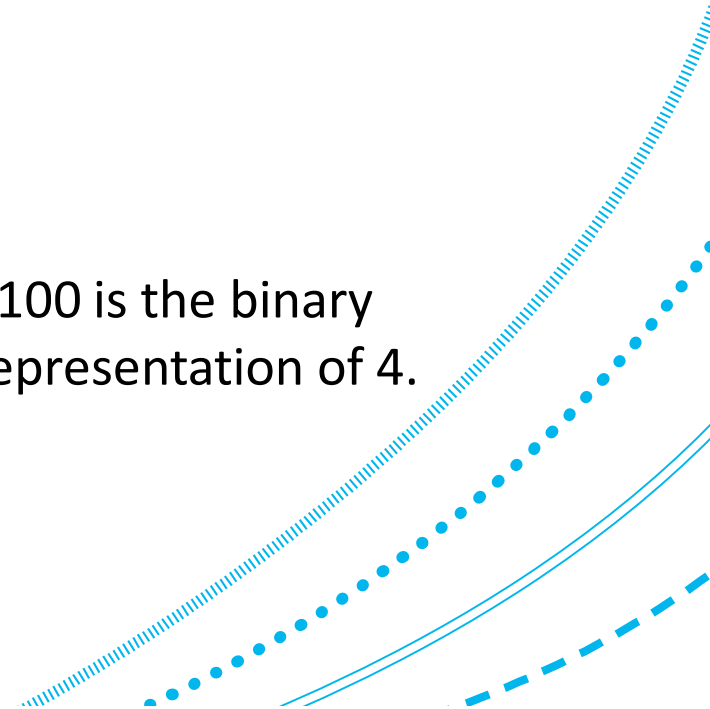
# How to obtain a two's complement representation by hand

1. Negative numbers are usually represented using a two's complement representation.
2. For example, -4

Calculation	Remainder
4 / 2	
2 / 2	0
2 / 1	0
	1



0100 is the binary representation of 4.



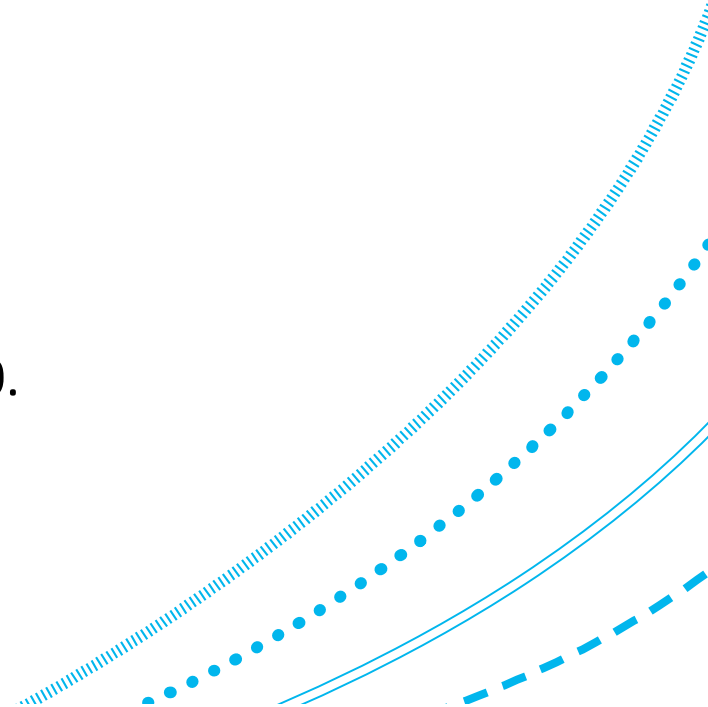


# How to obtain a two's complement representation by hand (cont'd)

1. Represent the binary representation of 4 using a byte-representation (1 byte = 8-bits), so 0100 in byte-representation is 0000 0100.
2. Flip each bit in the byte-representation, so 0000 0100 becomes 1111 1011.
3. Add one to the complement representation, so

$$\begin{array}{r} 1111\ 1011 \\ + \quad \quad 1 \\ \hline 1111\ 1100 \end{array}$$

4. The two's complement representation of -4 is 1111 1100.



## XOR Example 2

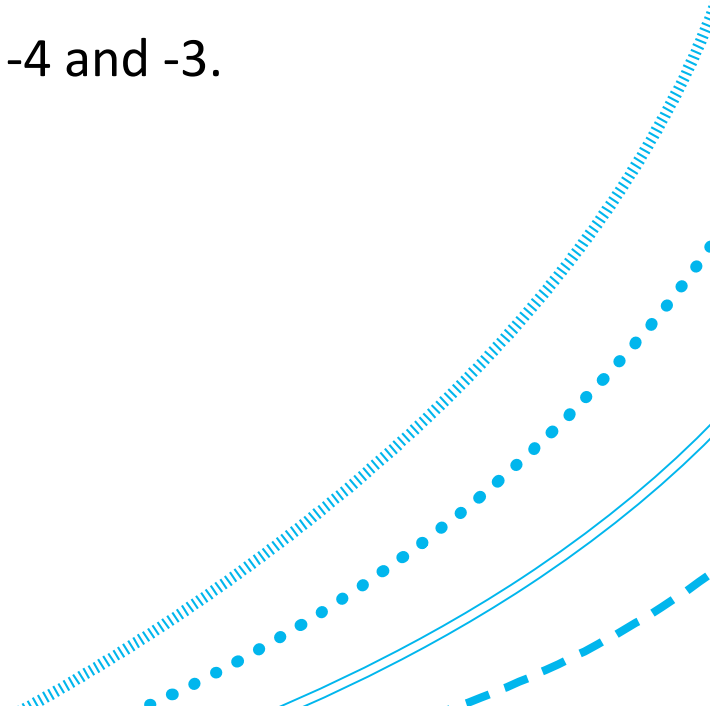
1.  $-4 \oplus -3$
2. Negative numbers are usually represented using a two's complement representation.
3. First step: obtain a two's complement representation of -4 and -3.

-4: 1111 1100

-3: 1111 1101

4. Second step: perform the XOR.

$$\begin{array}{r} 1111\ 1100 \\ \oplus \underline{1111\ 1101} \\ 0000\ 0001 \end{array}$$



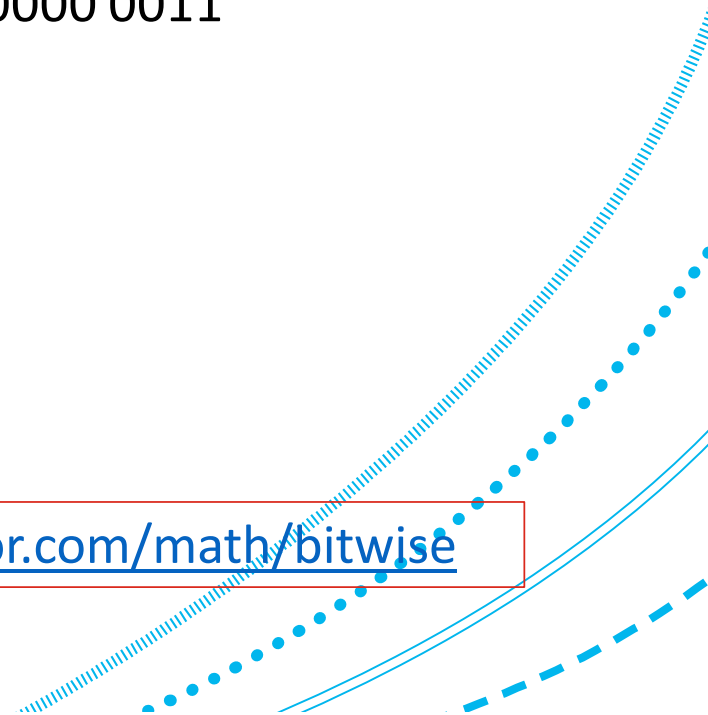
## XOR Example 3

1.  $-4 \oplus 3$
2. First step: represent -4 using two's complement representation, 1111 1100
3. Second step: represent 3 using its byte representation, 0000 0011
4. Third step: perform the XOR

$$\begin{array}{r} 1111\ 1100 \\ \oplus \underline{0000\ 0011} \\ 1111\ 1111 \end{array}$$

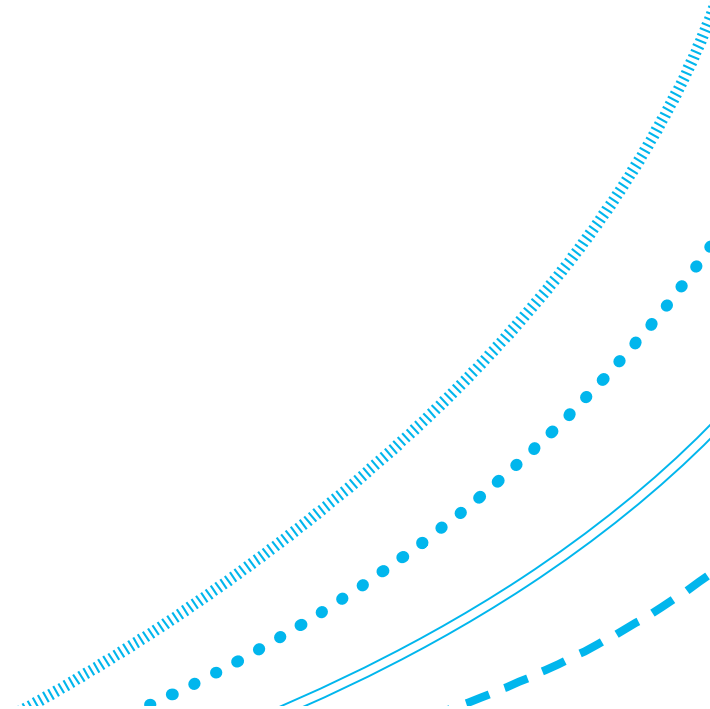
5. 1111 1111 is unsigned decimal result 255.

An online bitwise calculator, <https://www.omnicalculator.com/math/bitwise>



# Cryptographic modes

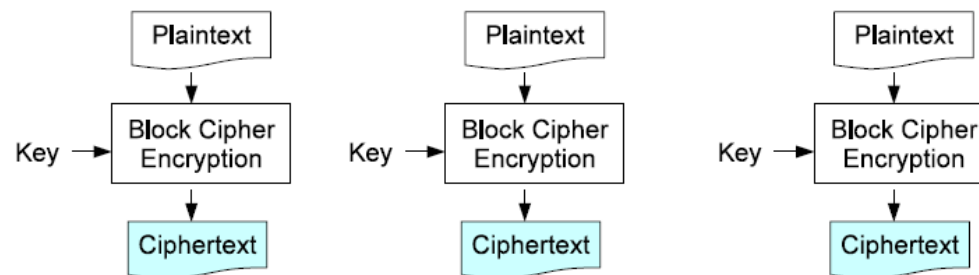
1. A block cipher encrypts a plain text in fixed-size  $n$ -bit blocks (often  $n = 64$ )
2. For messages exceeding  $n$  bit we can use four different modes of operation:
  1. ECB: Electronic Code Block.
  2. CBC: Cipher-Block Chaining.
  3. CFB: Cipher FeedBack.
  4. OFB: Output FeedBack.



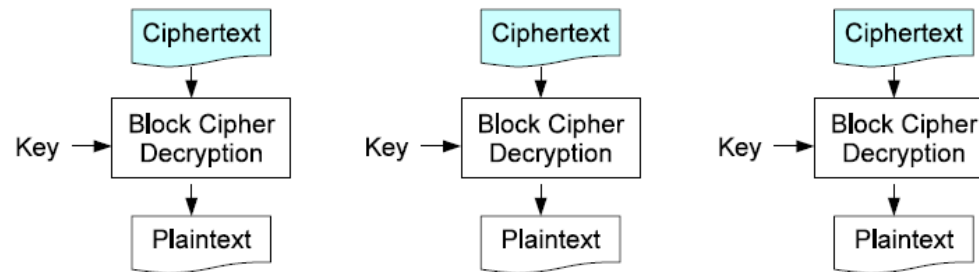
# Electronic code block

1. The message is divided into blocks and each block is encrypted separately:

$$c_j = E_k(m_j)$$



Electronic Codebook (ECB) mode Encryption



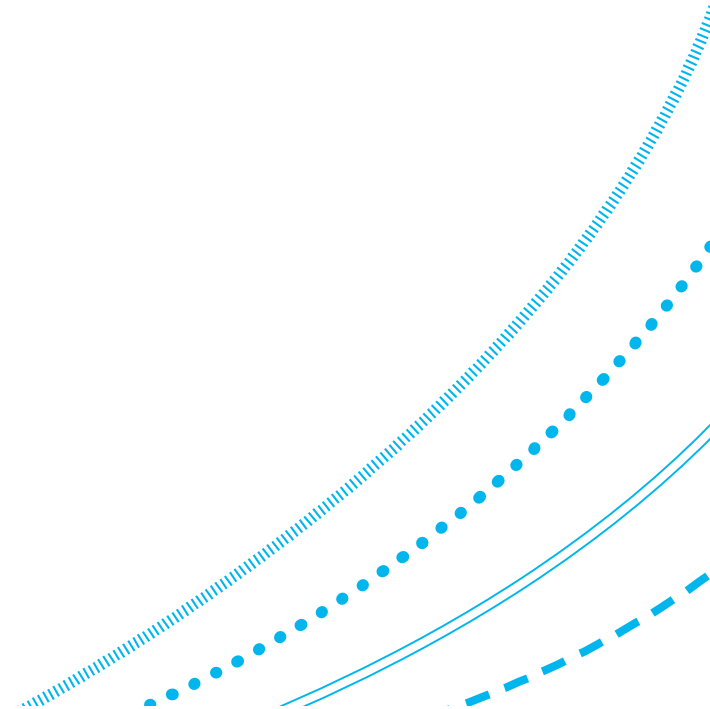
Electronic Codebook (ECB) mode Decryption

# Cipher block-chaining

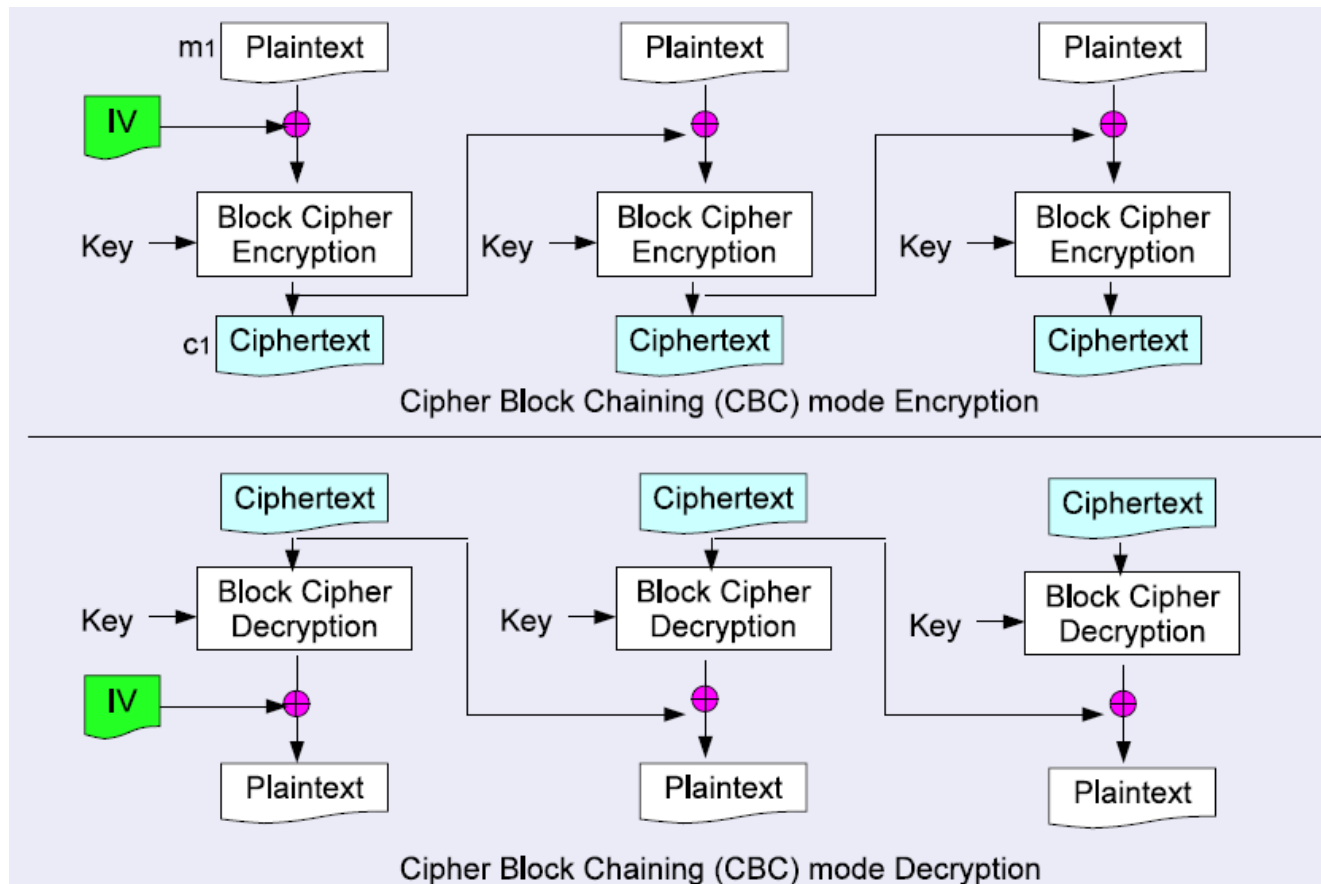
1. A plain text block is XORed with the previous cipher text block before encryption.
2. The first plain text block is XORed with an Initializing Vector IV:

$$c_1 = E_k(m_1 \oplus IV)$$

3.  $c_j = E_k(m_j \oplus c_{j-1})$

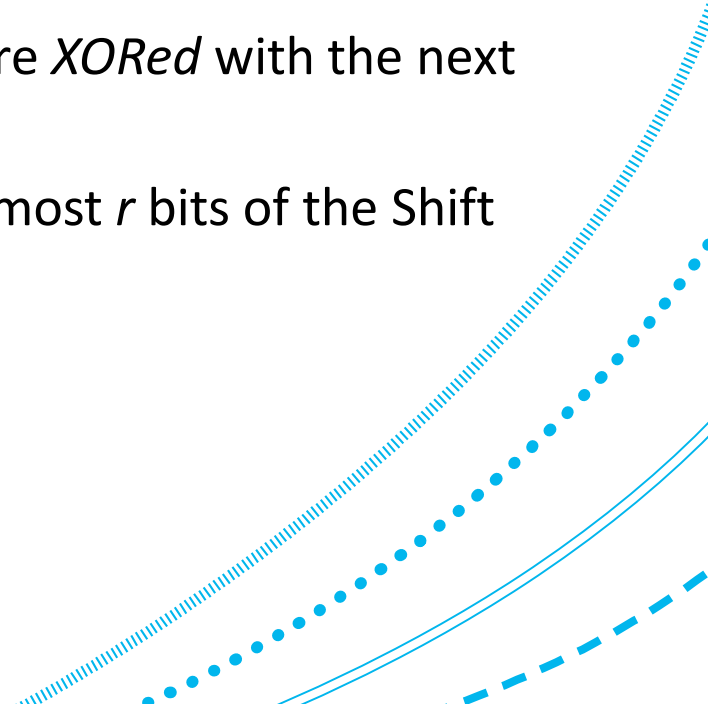


# Cipher block-chaining (cont'd)



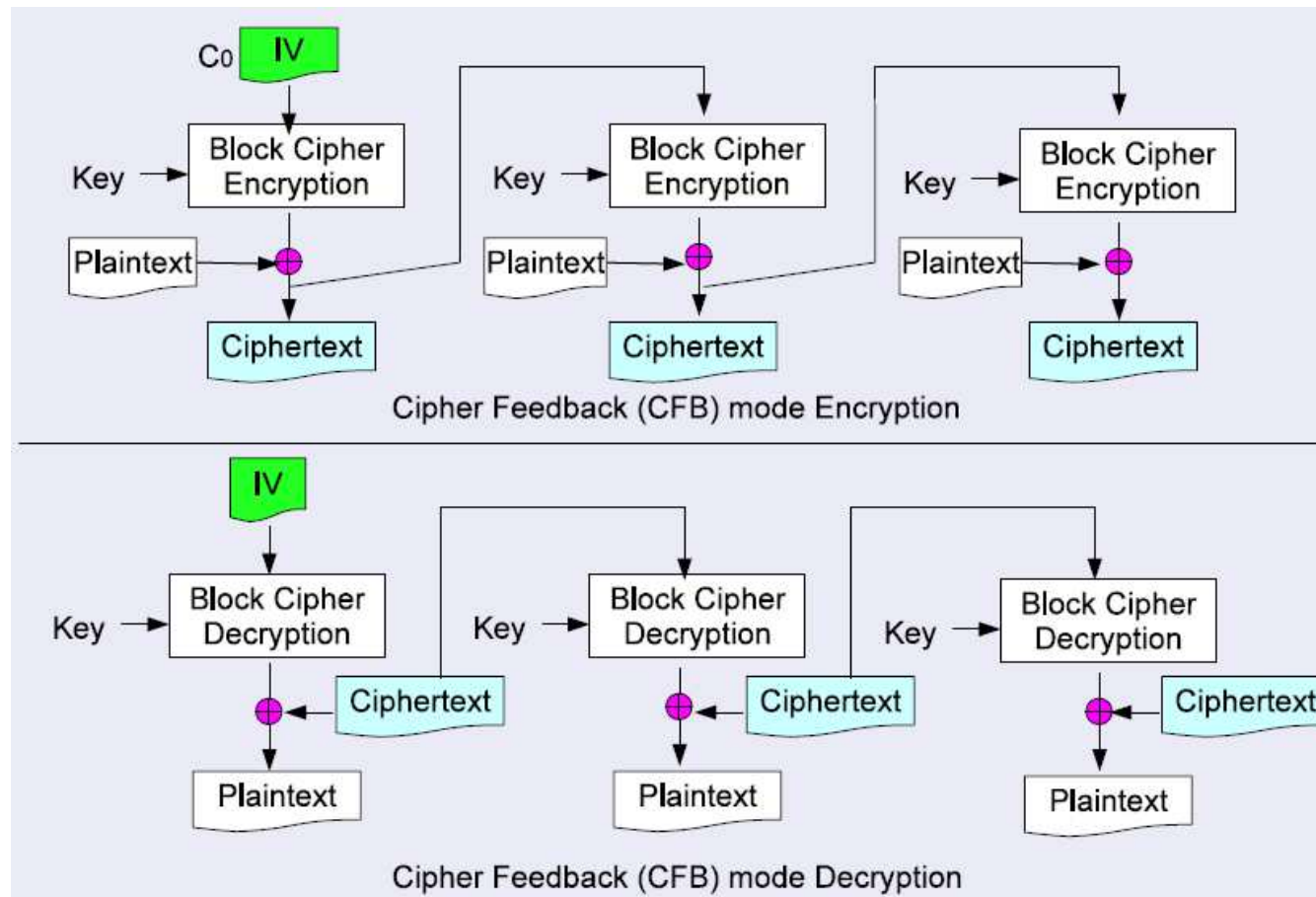
# Cipher FeedBack

1. Plain text is encrypted in blocks of size  $r$  ( $r < n$ ).
2. The  $n$ -bit Shift Register (initially IV) is encrypted into an intermediate cipher text.
3. The left-most  $r$  bits of the intermediate encrypted text are *XORed* with the next  $r$  bits of the plain text to obtain  $r$  bits of cipher text.
4. The  $r$  bits of the final cipher text are moved to the right-most  $r$  bits of the Shift Register and its  $r$  left-most bits are discarded.
5.  $C_i = E_k(C_{i-1}) \oplus P_i$ ;  $P_i = E_k(C_{i-1}) \oplus C_i$ ;  $C_0 = IV$



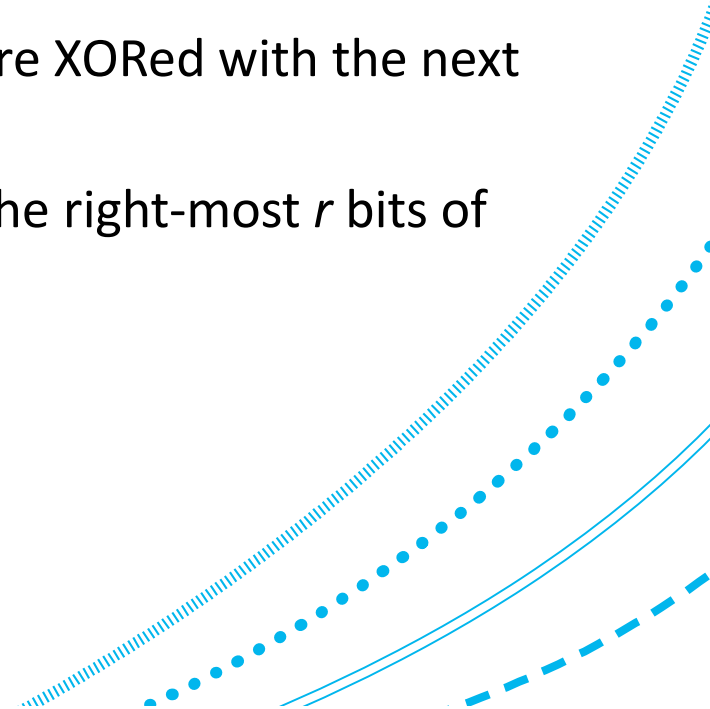


# Cipher FeedBack (cont'd)



# Output FeedBack

1. Plain text is encrypted in blocks of size  $r$  ( $r < n$ );
2. The  $n$ -bit Shift Register (initially IV) is encrypted into an intermediate cipher text;
3. The left-most  $r$  bits of the intermediate encrypted text are XORed with the next  $r$  bits of the plain text to obtain  $r$  bits of cipher text;
4. The  $r$  bits of the intermediate cipher text are moved to the right-most  $r$  bits of the Shift Register and its  $r$  left-most bits are discarded
5.  $C_i = P_i \oplus O_i$ ;  $P_i = C_i \oplus O_i$ ;  $O_i = E_k(O_{i-1})$ ;  $O_0 = IV$



# Output FeedBack (cont'd)

