Block1

EBU6010

Cryptography and Cyber Security

September 2023

Dr Yasir Alfadhl BEng(Hons.) PhD FHEA SMIEEE

yasir.alfadhl@qmul.ac.uk

School of Electronic Engineering & Computer Science

Queen Mary
University of London

Lecturers

Na Yao

Yasir Alfadhl

Course aims and objectives

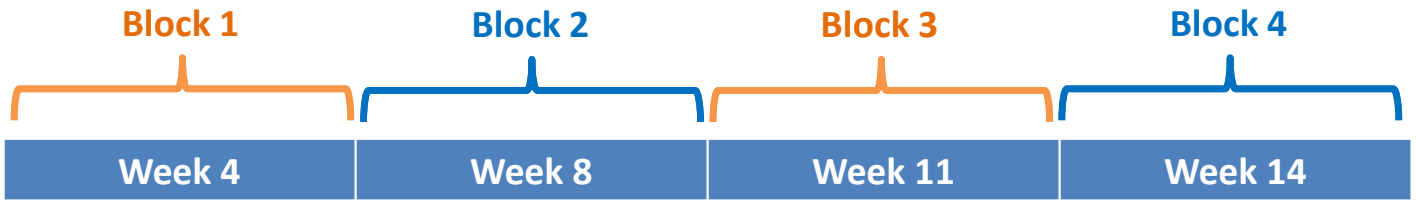
The course aims to introduce students to the principles and practice of cryptography and authentication used for network security.

Course Aim & Objectives

- **Main topics to be covered:**
  - **Security and Cryptography**
    - Introduction to security and Cryptography
    - Conventional Encryption
    - Public Key Cryptography
  - **Authentication**
    - Digital Signatures
    - Authentication Protocols
  - **Network Security**
    - Authentication applications
    - Web Security

Teaching Plan

- Dr Yasir Alfadhl (Blocks 1 & 3)
- Dr Na Yao (Blocks 2 & 4)



Teaching Plan – Block 1

- On Monday, Tuesday, Wednesday, and Thursday we will have lectures, during which we will deliver new material. It is important that you attend these to avoid falling behind.
- On Wednesday and/or Friday we will work through tutorial exercises to reinforce your understanding of the material.
- On Friday there will be a class test (CT)

Web Page

- **Course web page:**
  - qmplus.qmul.ac.uk/

- **Information provided:**
  - Lecture slides
  - Tutorial sheets
  - Information about the coursework
  - Deadlines
  - Etc.

- Regular Updates: Corrections, Cancellations, etc.

Recommended Books

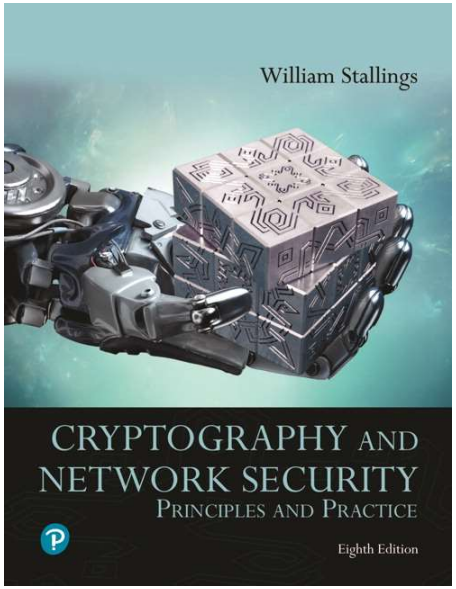
- Cryptography and Network Security: Principles and Practice, 8th Edition (2020)

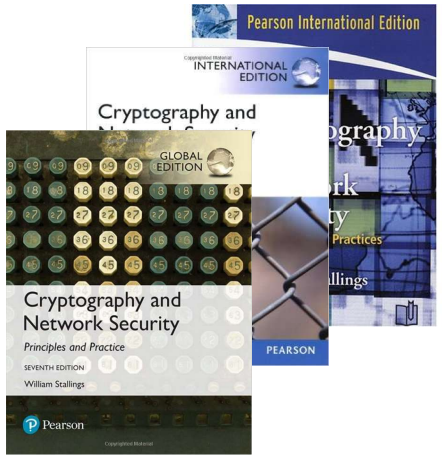
  W. Stallings, Prentice Hall

  Earlier editions are also ok!

- Network Security Essentials

  W. Stallings, Prentice Hall





Assessment

- The coursework counts **20%** of the final mark.
- Random class tests counts to a total of **5%**.
- Written exam **75%**.

Plagiarism

Treated very seriously and could lead to FAIL marks for the coursework or the entire course!

Plagiarism includes:

- The use or presentation of the work of another person as your own work (or as part of your own work) without acknowledging the source.
- Submitting the work of someone else as your own
- Extensive copying from someone else's work in your own paper or report.

Attendance

- **It is important to attend all lectures:**
  - Typically, student's performance is related to attendance;
  - Recorded lectures to be accessed on time.
  - Lecture notes are NOTES, details are discussed in the lecture room.

Plan

- **Block 1:**
  - Introduction, Conventional Encryption, ..
- **Block 2:**
  - Public-Key Encryption, Authentication, ..
- **Block 3:**
  - Kerberos, IPSec, Firewalls, ..
- **Block 4:**
  - Email Security, Web Security, ..

Module Representatives

Tutorials and Class tests

- Class tests count towards the final mark
- Office Hour: See timetable
- Tutorial: See timetable

Scrambling (Securing) data..

Historical examples..

Scrambled data?

- Morse Code (1)[... -- ...] (2)[... --- ...]
- The Enigma machine: Was widely used by Nazi Germany; its cryptanalysis by the Allies provided vital Ultra intelligence.

Classical and medieval cryptography:

- Egyptian; Cryptography is found in non-standard hieroglyphs carved over 4500 years ago (attempts to intrigue/amuse literate onlookers).
- Mesopotamian: Clay tablets with encrypted valuable recipes. Later, Hebrew scholars made use of simple substitution ciphers (such as the Atbash cipher) beginning perhaps around 500-600 BC.
- Chinese: A substitution table by the strategist *Sun Tzu* (~500 B.C.) gave a code comprising 40 elements, assigned to 40 characters of a poem.
- Greek: Scytale transposition cipher (by Spartan military ~650 B.C.).
- Romans: The Caesar cipher and its variations ~100 B.C.
- Arabic: ~ 750 A.D. Al-Kindi has published a book entitled "Manuscript for the Deciphering Cryptographic Messages" ("Risalah Fi Istikhraj Al-Mu'amma") -- Cryptanalysis techniques, classification of ciphers and described the use of several statistical techniques for cryptanalysis.

The Enigma machine
Source:
http://en.wikipedia.org/wiki/History_of_cryptography

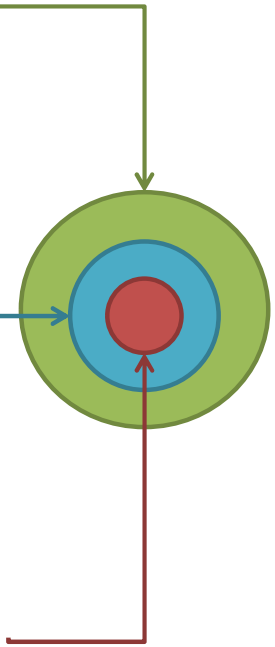
A Scytale, an early Greek device for encryption.
Source:
http://en.wikipedia.org/wiki/Scytale

Introduction

- The transmission of data over networks is the core of almost all technologies.
- Most of the transmitted data may contain sensitive information (bank details, personal records, technical info, etc).
- **How could we ‘protect’ such information?**
- Other applications from crypto algorithms?

Information Systems Security

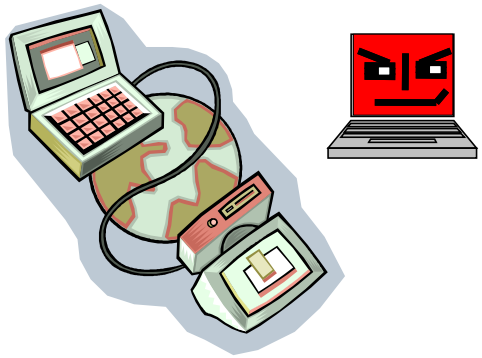
- **Informal**
  - Educating and training the members of organisation
- **Formal**
  - Data management or security rules
  - Management of personnel
- **Technical (Technology Based):**
  - Smart security cards, Ciphers, etc.



Internet Security

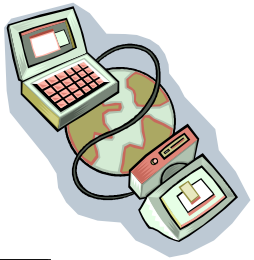
- **A security system is typically introduced to:**

Deter, Prevent, Detect and Correct security violations of data transmission.



Security Architecture

- **Security Attacks**
  - Actions involving the compromise of security info.

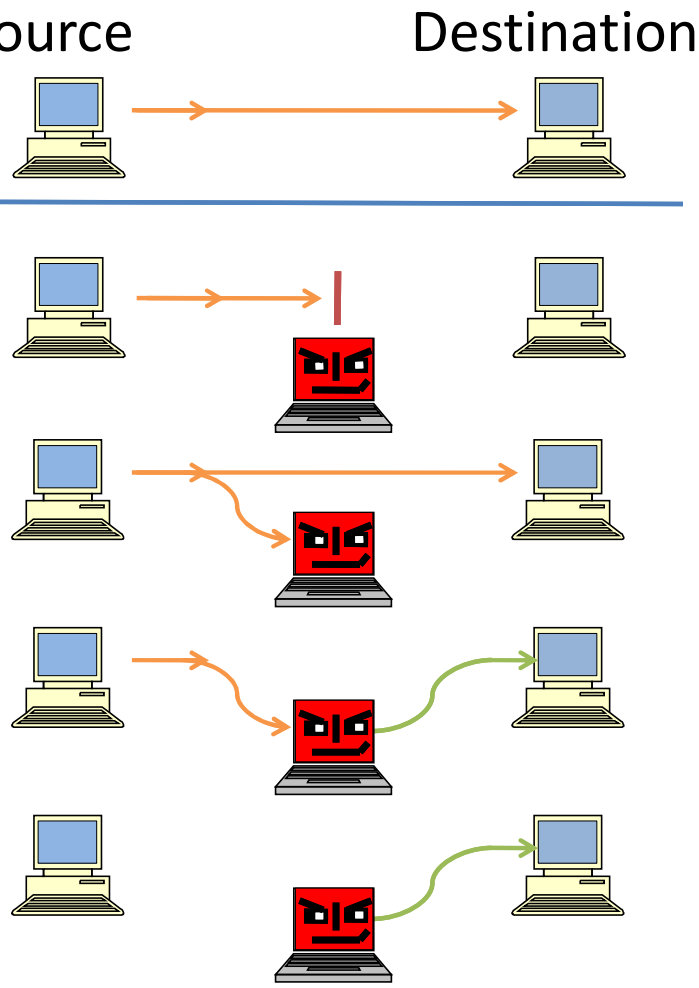


- **Security Mechanisms**
  - Detection, prevention and recovery from attacks.

- **Security Services**
  - Processes which improves security and protects from attacks.

Terminologies of Security Attacks

- Normal Flow
- Interruption
- Interception
- Modification
- Fabrication



Security Attacks

- **Passive Attacks**
  - Release of message contents — Intercept info
  - Traffic Analysis — Analysis of traffic volume data

- **Active Attacks**
  - Masquerade — Capture and replay of valid authentication sequence
  - Replay — Re-use of observed data to produce an unauthorised effect.
  - Modification of message contents — Or part of it.
  - Denial of Service — Inhibits the normal use of the network. E.g. Network flooding or redirection of traffic.

Security Mechanisms

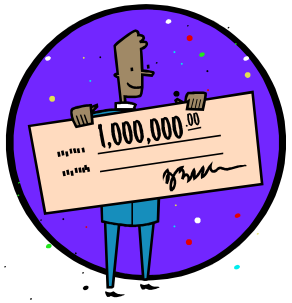
- There is no single mechanism to provide information security.
- However, the element that underlies most of the security mechanisms is the use of ‘Cryptographic Techniques’.
- **Cryptography** is the art of secret writing, is the process of converting information, such as this slide, that can be read by most, into a secret code, that can only be read by those who are party to the secret.

Cryptography
Originates from the Greek Krypto 'hidden' and Grafo 'written'.

Security Services

- **Authentication**
  - Assurance of valid users and logical connections.
- **Access Control**
  - Prevention of unauthorised used of recourses.
- **Data Confidentiality**
  - Protection from unauthorised disclosures
- **Data Integrity**
  - Assurance of valid/unchanged data.
- **Non-repudiation**
  - Protection against denial from either party.
- **Data Availability**

Bank Cheque Example

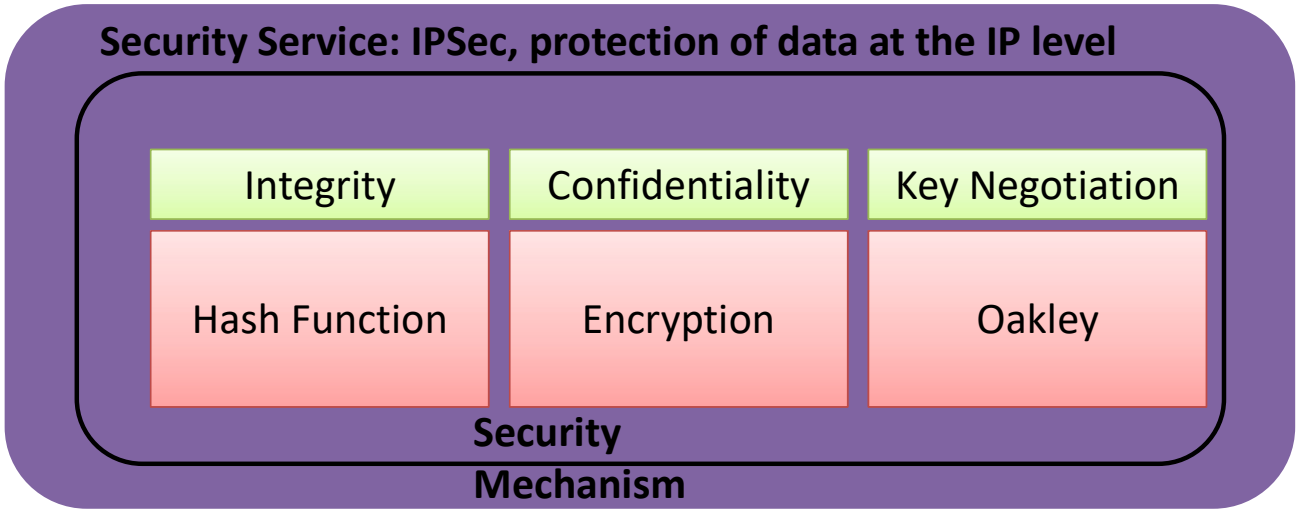


Terminology

	Term	Description
Security	**Plaintext**	Original message
	Encryption	Encoding the message to hide its contents
	Ciphertext	Encrypted message
	Decryption	Retrieving the plaintext from ciphertext
	Key	Is used by the encryption and decryption. The decryption can be performed only by knowing the proper key.
Mechanism	**Encryption**	Confidentiality, authentication, integrity protection.
	Check/Hash algorithms	Integrity protection, authentication
	Digital signatures	Authentication, integrity protection, non-repudiation.
Services	**Access control**	Unauthorised user
	Confidentiality	Disclosure of unauthorised identities
	Integrity	Unauthorised data alterations
	Non-repudation	Originator of communications , later denying it
	Authentication	Assurance of someone's identity

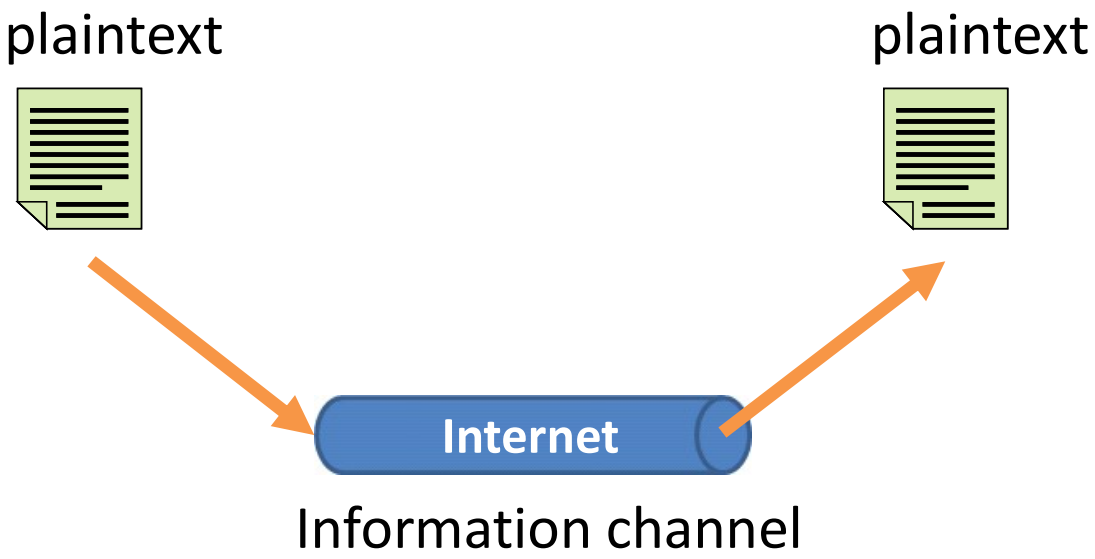
Security Services

- One or more security mechanisms are combined to provide a security service.
  - IPSec, protection of the data at the IP level.
  - Ensures adequate security of the system resources and data transfer.



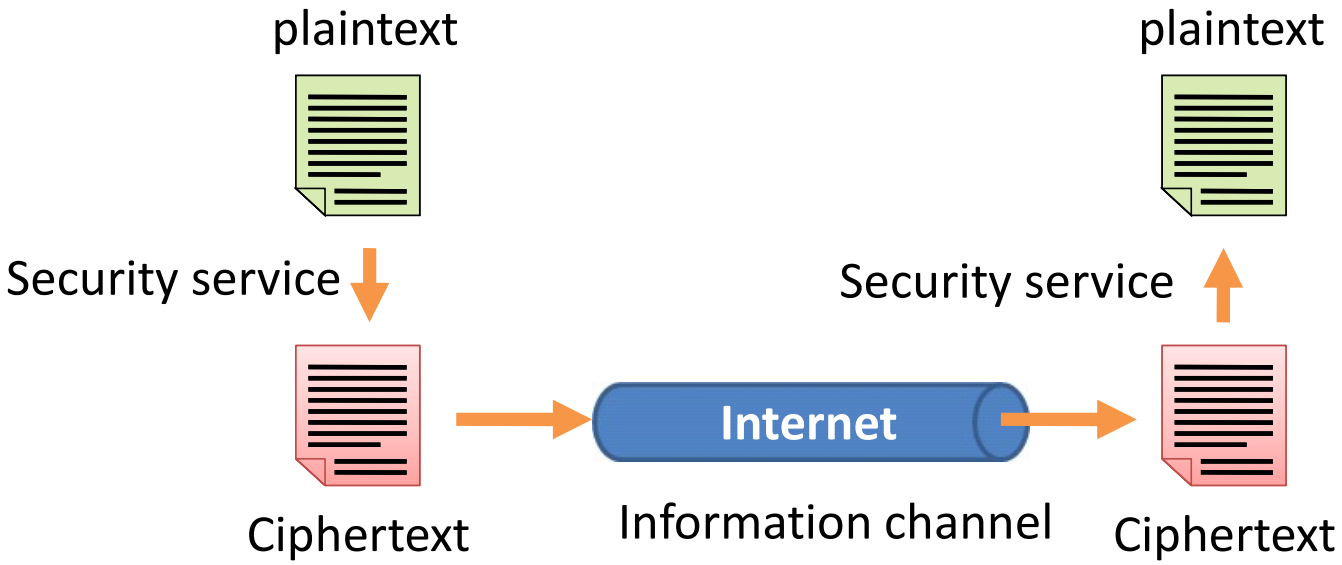
A model of Internet security

- Insecure channel



A model of Internet security

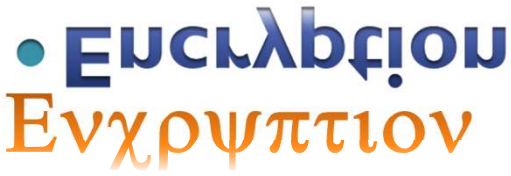
- Secure channel
  - Trusted third party (Distribution of secret information)



Summary

- The course focuses mainly on Internet Security
- Security is assessed by the attacks, services and mechanisms
- Several security mechanisms can be combined to provide a 'Security Service'.
- The main security mechanisms used in the internet are based on cryptographic techniques.
- The terminology in encryption is: plaintext, ciphertext, encryption, decryption and key.
- Different security mechanisms protect against different attacks.

Encryption



AXafaaxvggavgxfgdaaf

A simple Example

- **Caesar Code**

1. Write the alphabet

2. Write the alphabets again underneath, but starting from the letter '**d**', if you run out of letters start again with '**a**' etc.

3. From the original message substitute the original letter for the shift letter.

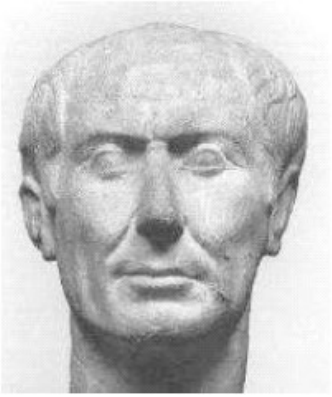
Caesar Cipher

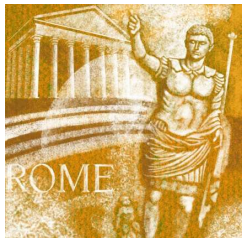
- **Key:** new letter = old letter +3

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
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

- **Example:**

  Julius Caesar ➔ Mxolxv Fdhvdu





Mathematical expression of Caesar Cipher

- Assign a number to each letter, a=0, b=1, ... z=25

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Key= number of spaces forward in alphabet from plaintext letter

Note: **m** = plaintext, **k** = secret key, **c** = ciphertext

$c = m + k = m + 3$ or, more formally, $(m + 3) \bmod 26$

mod 26 refers to the modulo. Modulo is the operation of finding the Remainder when you divide two numbers. So, c cannot be > 26

E.g. letter J becomes letter M i.e. [9(letter J) + 3(Key)]mod26 = 12(letter M)

Try to solve this..

- Caesar wants to arrange a secret meeting with Marc Anthony, either at the Tiber (the river) or at the Coliseum (the arena).
- He send the ciphertext **EVIRE**.
- However, Antony does not know the key, so he tries all possibilities.
- Where will he meet Caesar?

*Source: Introduction to Cryptography with Coding Theory, W. Trappe and L. Washington, Pearson, Prentice Hall, 2006.

Caesar Cipher

- Caesar cipher, is a **stream cipher**, that uses simple **mono-alphabetic substitution**.

- It is very simple to break (his successor Augustus used a one shift key, perhaps he could not safely count to three).

Polyalphabetic Substitution:
Vigenère Cipher

- There are stream ciphers that use **poly-alphabetic** substitution. An example is the 'Vigenère Cipher'

1. Identify letters with numbers, a=0, b=1, ..., z=25
2. The secret key is a sequence of letters, e.g. a word.
3. Encrypt by adding the plaintext letter to a key letter using rotation.

Example: Vigenère Cipher

- **Plaintext**: my password is tomato
- **Key**: stream

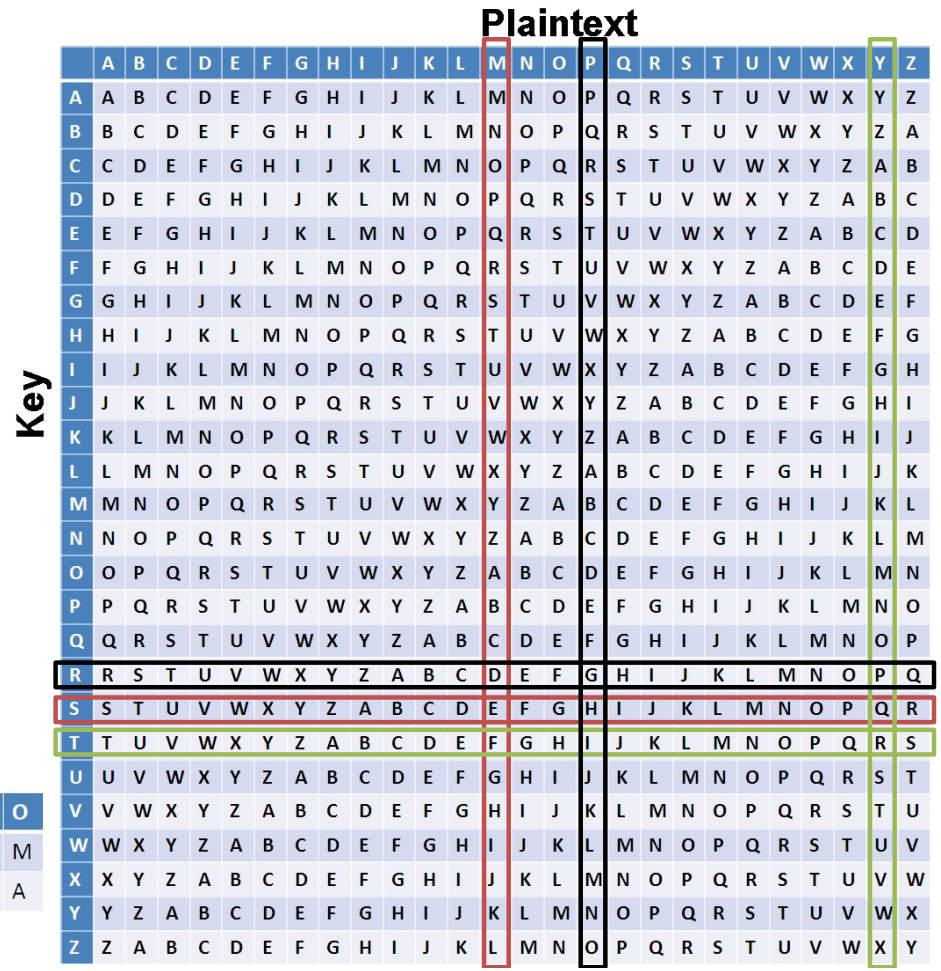
PlainText	M	Y	P	A	S	S	W	O	R	D	I	S	T	O	M	A	T	O
Key	S	T	R	E	A	M	S	T	R	E	A	M	S	T	R	E	A	M
Ciphertext	E	R	G	E	S	E	O	H	I	H	I	E	I	H	D	E	T	A

Example: Vigenère Cipher

How does it work?

- The first letter in the plaintext is ‘M’ and the first letter in the key is ‘S’
- Move to column ‘M’ and row ‘S’
- And that is cipher ‘E’
- Repeat the process..

PlainText	M	Y	P	A	S	S	W	O	R	D	I	S	T	O	M	A	T	O
Key	S	T	R	E	A	M	S	T	R	E	A	M	S	T	R	E	A	M
Ciphertext	E	R	G	E	S	E	O	H	I	H	I	E	I	H	D	E	T	A



Example: Vigenère Cipher

- Using numbers:
  - M ➔ 12 plaintext
  - S ➔ 18 key

- Encryption:
  - (12+18)mod26 = (30)mod26 = 26 + 4 = 4
  - 4 ➔ E Ciphertext

* Used in Enigma.

Example: Rotor Encryption

- HELLO → EROFW

26 Alphabets

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
21	22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
13	24	6	25	2	18	23	12	9	17	5	11	4	22	7	16	8	20	26	14	10	19	1	15	3	21
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

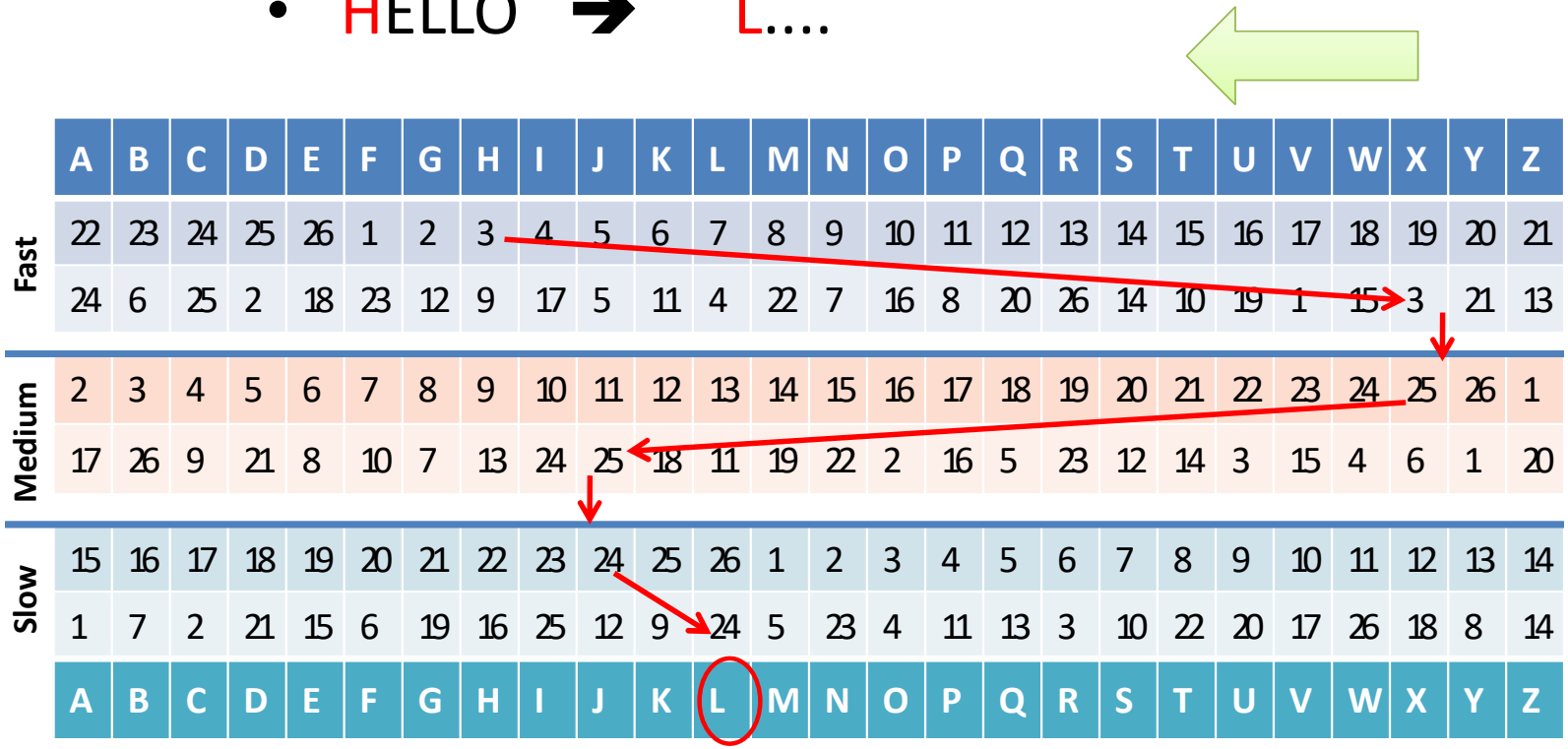
.. and the second letter ..

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
22	23	24	25	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
24	6	25	2	18	23	12	9	17	5	11	4	22	7	16	8	20	26	14	10	19	1	15	3	21	13
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

Example: Rotor Encryption

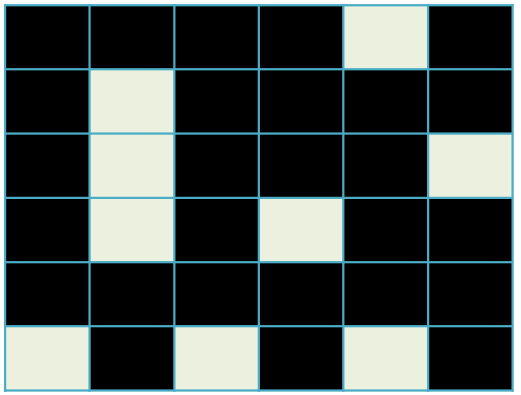
If 3 rotors → 26^3 = 11,567
If 5 rotors → 26^5 = 11,881,376 Alphabets

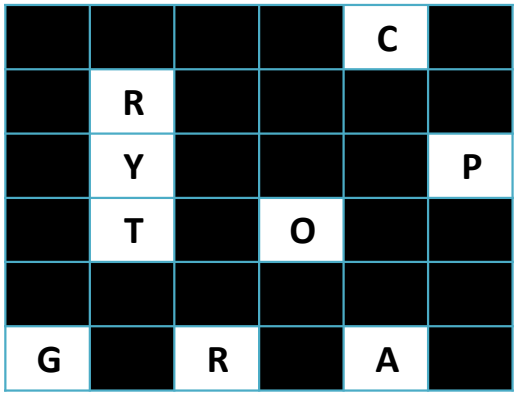
- HELLO → L....



Transposition: The Grille

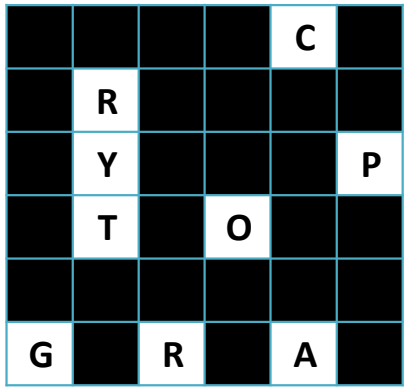
P	R	T	T	C	O
W	R	H	Y	I	R
S	Y	F	I	S	P
E	T	T	O	C	T
H	I	N	G	R	E
G	E	R	A	A	.

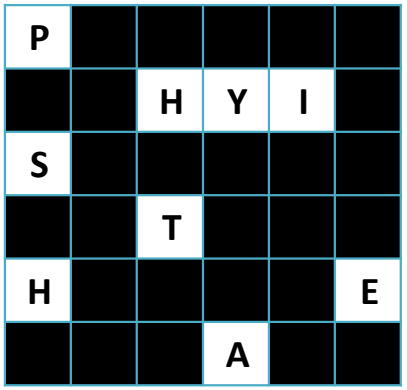


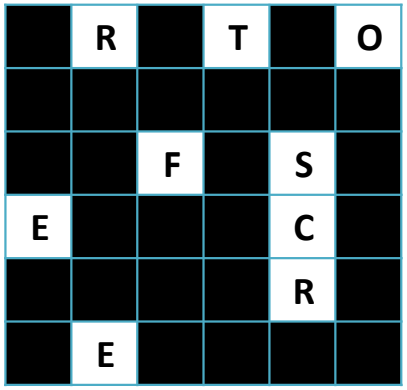


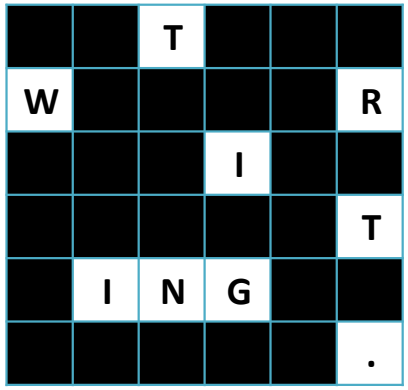
Transposition: The Grille

P	R	T	T	C	O
W	R	H	Y	I	R
S	Y	F	I	S	P
E	T	T	O	C	T
H	I	N	G	R	E
G	E	R	A	A	.







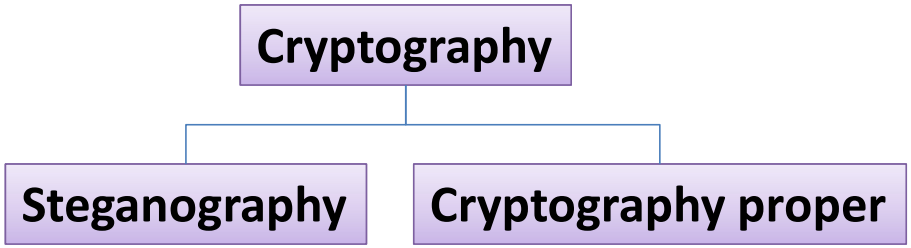


- The images on this slide illustrate the effect of performing 4 consecutive clockwise rotations ((through 90 degs.) of the Grille

Classification of Cryptographic Systems

Classification of Steganography & Cryptographic Methods

- **Steganography (covert secret writing)**
  - covert writing, is where it is not evident that there is a secret message
- **Cryptograph proper (overt secret writing)**
  - Overt writing, is evident that there is a secret message.



*Source: Decrypted Secrets, F.L.Bauer, Springer, 3rd Edn. 2002.

Classification of Cryptographic Systems

1. **The type of operations used for transforming plaintext to Ciphertext**
2. **The number of keys used**
3. **The way in which the plaintext is processed**

In current encryption techniques the security depends on the secrecy of the algorithm.

Classification of Cryptographic Systems

- **Types of operations:**

1. **Substitution:** Each element of the plaintext is mapped into another element. (element = bit, letter, group of letters ...)

2. **Transposition:** Each element of plaintext is rearranged.

Method	Example	Explained..
Substitution	Caesar ➔ Mxolxv	Substitute one letter for another.
Transposition	Caesar ➔ raaCse	Change the order of the letters.

Diffusion and Confusion (Shannon):
No information is lost, and the operations are reversible.

Classification of Cryptographic Systems

The number of keys used:

- **Symmetric:** Sender and receiver use the same key.
  - This is known as '***conventional encryption***'.

Also known as 'Single-key' & 'Secret-key'

- **Asymmetric:** Sender and receiver each use a different key.
  - This is known as '*public-key encryption*'.

Also known as 'Two-key' encryption.

Classification of Cryptographic Systems

Process of the plaintext:

- **Stream Cipher**: Process one input element at a time.
- **Block Cipher**: Process a block of elements at a time.

Stream Ciphers

- **Notation:**
  - **m** = plaintext, **k** = secret key, **c** = ciphertext
  - **e** = encryption function, **d** = decryption function
- **Encryption:**
  - $c = e_k(m)$
  - $c_i = m_i \oplus s_i$
  - Where: i = 0,1,...., s_i = key bit stream, and $\oplus$ is the XOR function
- **Decryption:**
  - $m = d_k(c)$
  - $m_i = c_i \oplus s_i$

Input	Output
$0 \oplus 0$	0
$0 \oplus 1$	1
$1 \oplus 0$	1
$1 \oplus 1$	0

Stream Ciphers

- **Example**

ASCII

A	01000001
B	01000010
C	01000011
D	01000100
:	**:**

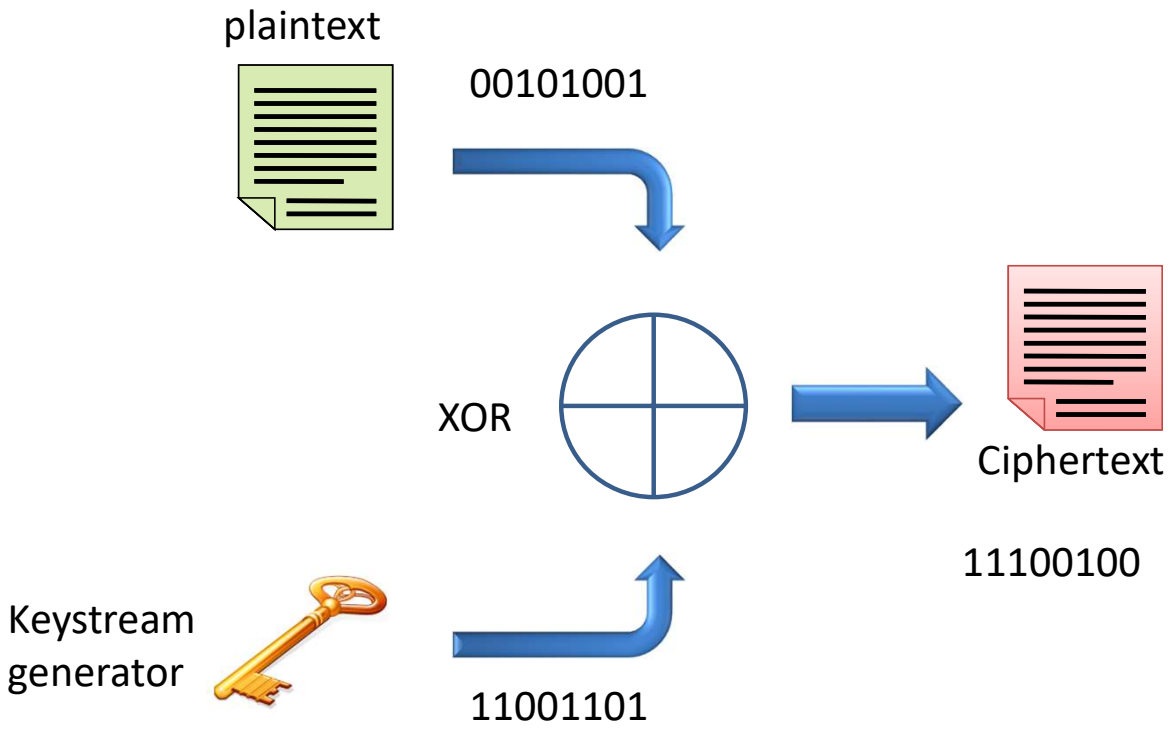
XOR

Input	Output
$0 \oplus 0$	0
$0 \oplus 1$	1
$1 \oplus 0$	1
$1 \oplus 1$	0

plaintext	1	0	1	1	0	1	1	1	0	..
key	0	0	1	1	1	0	1	0	1	..
ciphertext	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**..**

Stream Ciphers

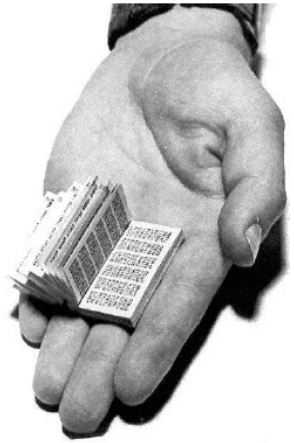
- Notice that the composition of two XOR's is identical to the original data.



Stream Ciphers

- **Encryption can be very fast**
- **No error propagation, but ..**
  - No protection against message manipulation
  - It is easy to determine the key-stream if one knows the plaintext and ciphertext
  - If one bit, in either the message or key, is in error then the entire piece of cypher text will become corrupted

One Time Pad



- **There is a stream cipher that is <u>unbreakable</u>,**
  - This is known as the '***one time pad***'.
- **How it works?**
  - For each message use a new random key that is as long as the message.
  - Encryption output that has no statistical relationship to the plaintext.
- **Applications**
  - Secure media
- **Vulnerabilities**
  - The practical difficulty is how to transmit and protect the random key.
  - Message manipulation
  - Like other stream ciphers, easy to get wrong!

Example: One Time Pad

- An example using letters (instead of bits)
- The encryption is done using Vigenère cipher where the key is a random collection of letters. The key length is equal to the plaintext length.
- The first line is the plaintext, the second line is the key, the ciphertext is in green (See next slide).

One Time Pad (message manipulation)

- The following messages and keys produce the same ciphertext

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

Both produce

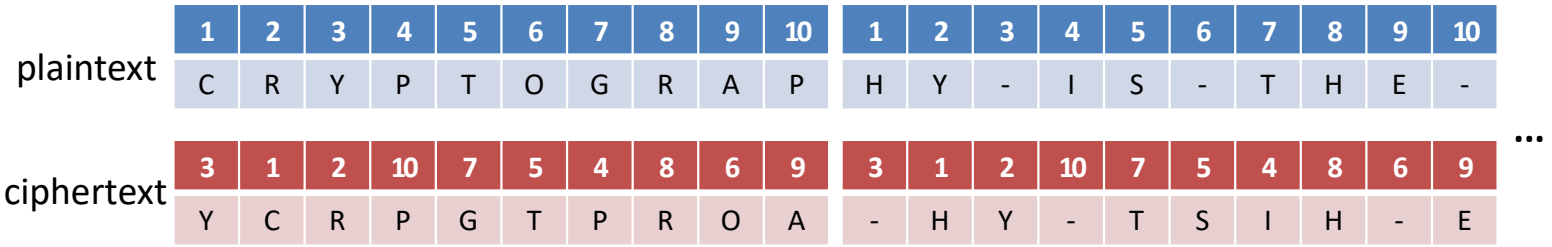
B	O	W	G	N	F	S	P	G	K	M	N	F	K	C	C	Y	C	F	Q	W	I	T	G	M	E	F	O	Q	N	V	E	O	E	Y	M

- If you only know the ciphertiext which one is the original text?

Simple Block Cipher

- Takes the letters and changes their order.
- **For example:**
  - Block size is 10 letters
  - Permutation: from {1,2,3,4,5,6,7,8,9,10} to {3,1,2,10,7,5,4,8,6,9}

Plaintext	Ciphertext
cryptography-is-the-art-of-secret-coding	ycrpgtproa-hy-tsih-etarc-o-sfetregdc-ion



Playfair square

* Used by US army in WWI and as late as WWII.

- Uses a 5X5 table
  - Contains a key word or phrase (without repeating letters)
  - Then filled with the remaining alphabets.
  - In order to fit the square, some systems omit the 'Q', and others combine the I&J in the same square

- Eg. "MY SECRET CODE IS" ➔MYSECRTODI

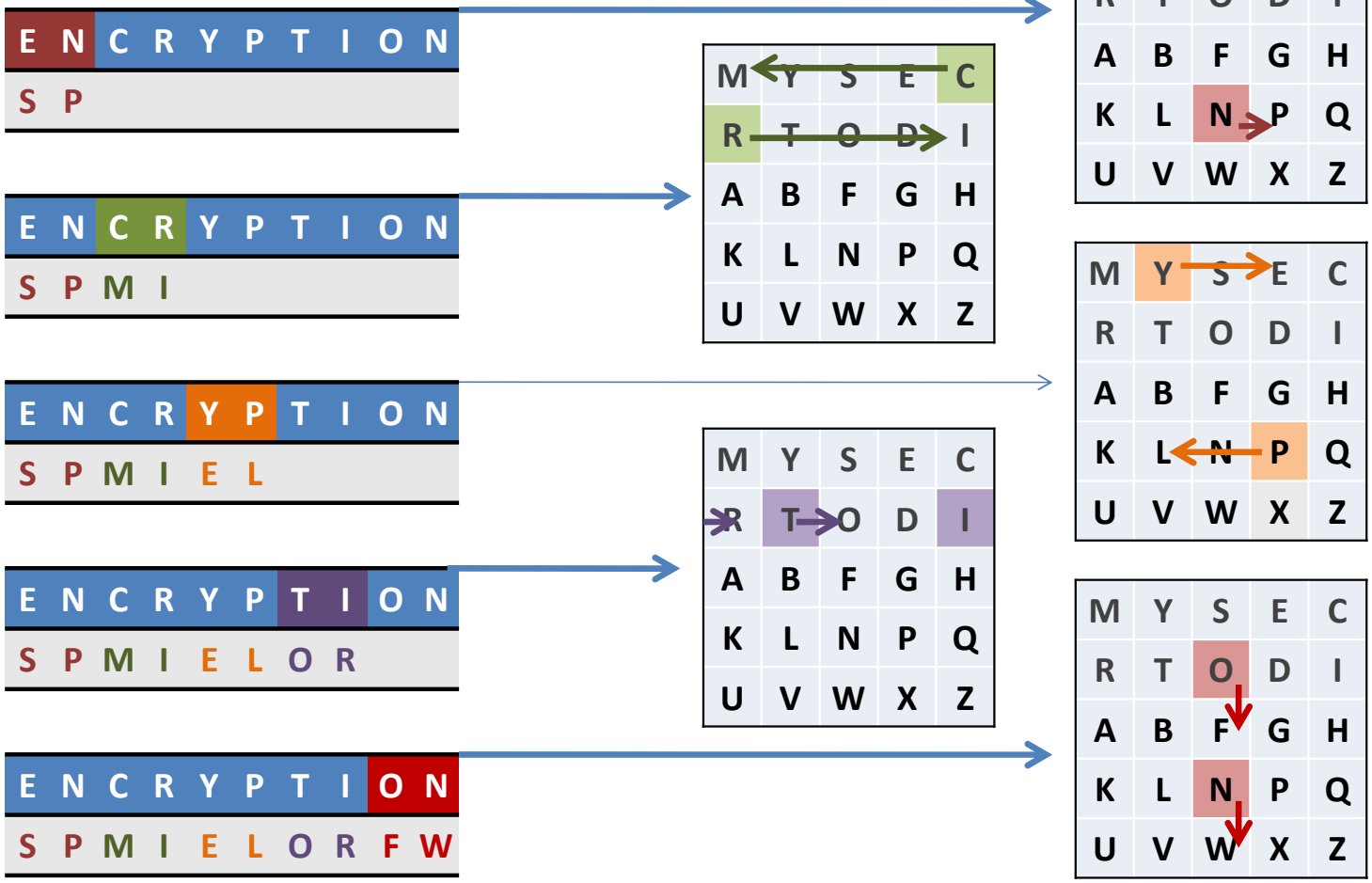
M	Y	S	E	C
R	T	O	D	I
A	B	F	G	H
K	L	N	P	Q
U	V	W	X	Z

Playfair square

- Break the message into groups of two letters and map them into the key table. The two letters of digraph are considered as opposite corners of a rectangle.
  - If the group consist of similar letters, insert a ‘Q’ or ‘X’.
  - If both letters are on the same row, replace them with their immediate right respectively (wrapping around)
  - If both letters are on the same column, replace them with the letters immediately below (wrapping around)
  - All other letters must be replaced by the other two corners of the formed rectangle (in the order they are placed).
- Decryption is achieved by inverting the process, with dropping any extra ‘X’ or ‘Q’ that don’t make sense!

E N C R Y P T I O N

Playfair: Example

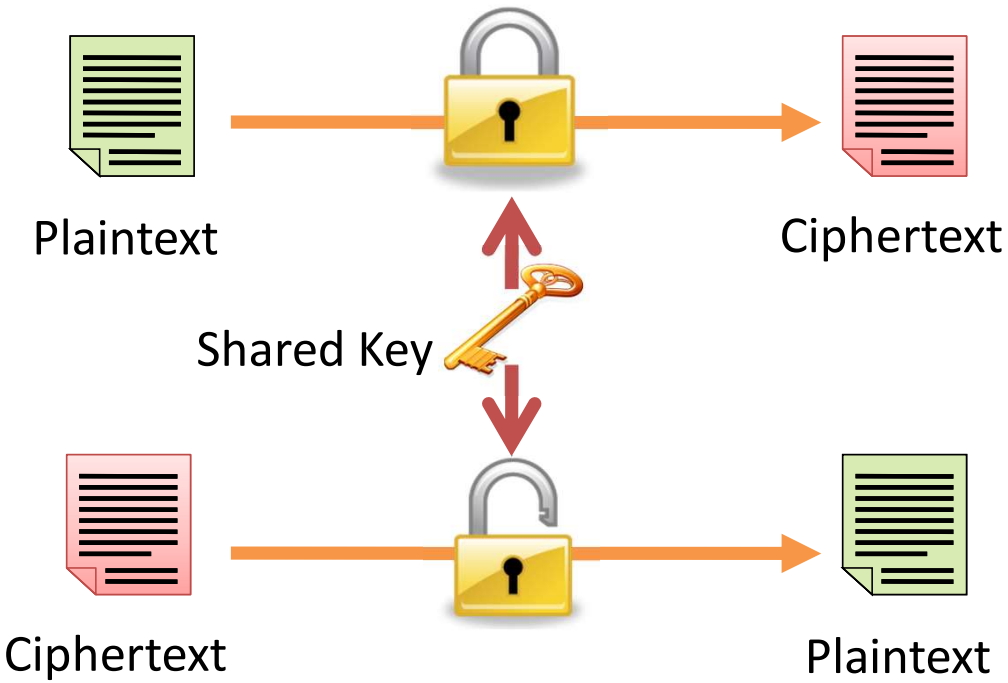


Security of conventional encryption

- Strong encryption algorithm
- Sender and receiver obtained the secret key in a secure fashion
- The key must be kept secure at all times

Encryption and the Key

- Encryption and decryption share the same key



Key Agreements

- In modern encryption the algorithms are public, the strength of the structure communication mechanism is based on the secrecy of the key.

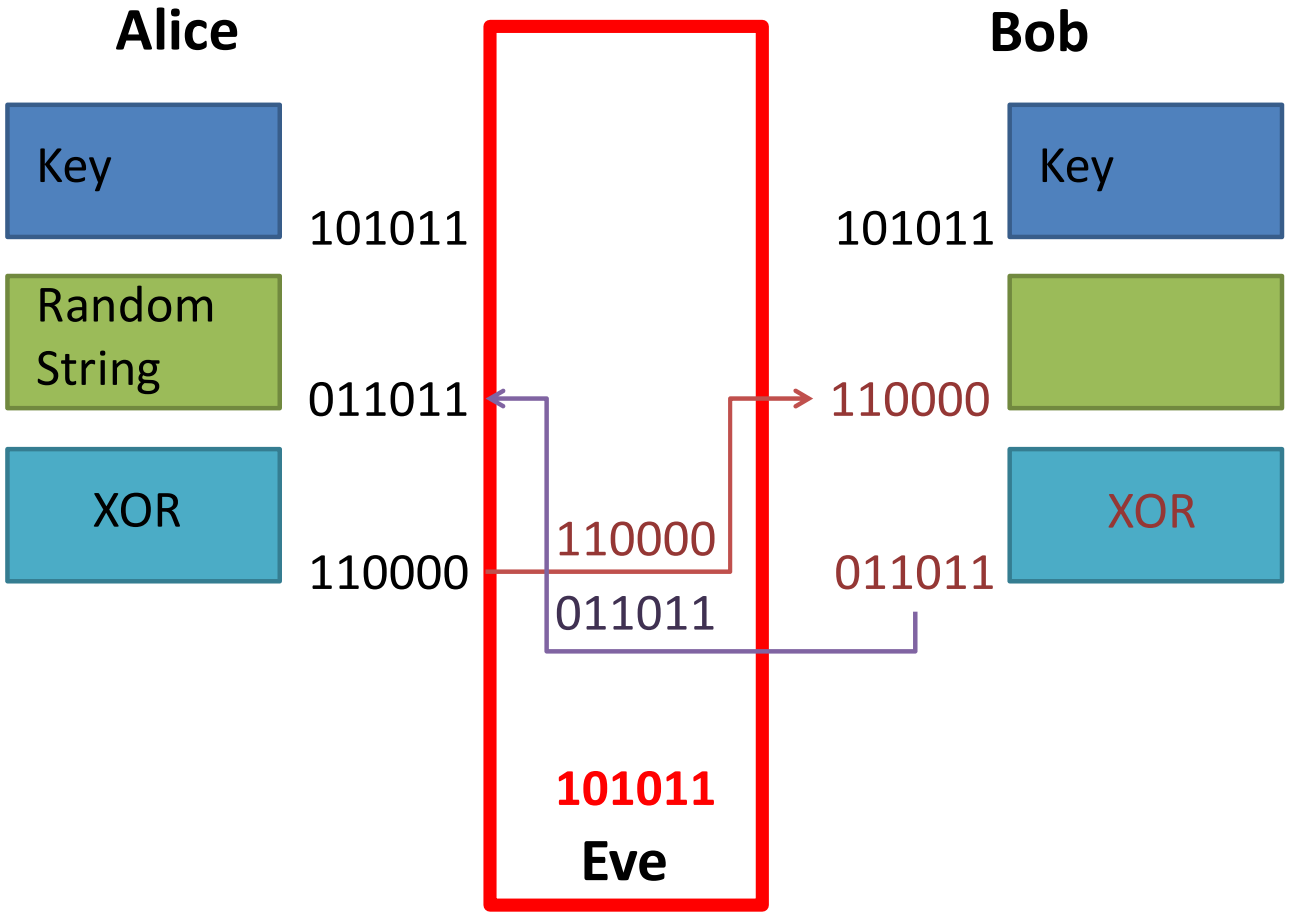
- Hence key agreement is a security mechanism that is of fundamental importance as it deals with agreement on shared secure channel to exchange conventional encryption key

- To exchange the keys used for encryption we need:
  - Agreement of shared key
  - Secure channel to exchange conventional key

Secure Key Exchange?

- **Is there a flaw in the following scheme to confirm that Alice and Bob are both in possession of the same secret key?** [Example from the course textbook]
  - Alice creates a random bit string the length of the key, XORs it with the key, and
  - Sends the result over the channel to Bob
  - Bob XORs the incoming block with the key (which should be the same as Alice's key) and Sends it back
  - Alice checks and if what she receives is her original random string, she has verified that Bob has the same secret key, yet neither of them has ever transmitted the key.

Secure Key Exchange?



Typical Attack Approaches

- **Cryptanalysis Attacks:**
  - The attacker relies on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext-ciphertext pairs.
  - The aim is to deduce a specific plaintext or the key being used.

- **Brute-force Attacks:**
  - The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained.
  - On average, the attacker succeeds after 50% of the trials.

Cryptanalysis

- **The process of attempting to discover the plaintext or key from the ciphertext.**

- **In general, an encryption algorithm, is designed to withstand an attack even when**
  - The ciphertext
  - The encryption algorithm
  - One or more plaintext-ciphertext pairs formed with a secret key are known

- **This is known as a known-plaintext attack.**

Cryptanalysis

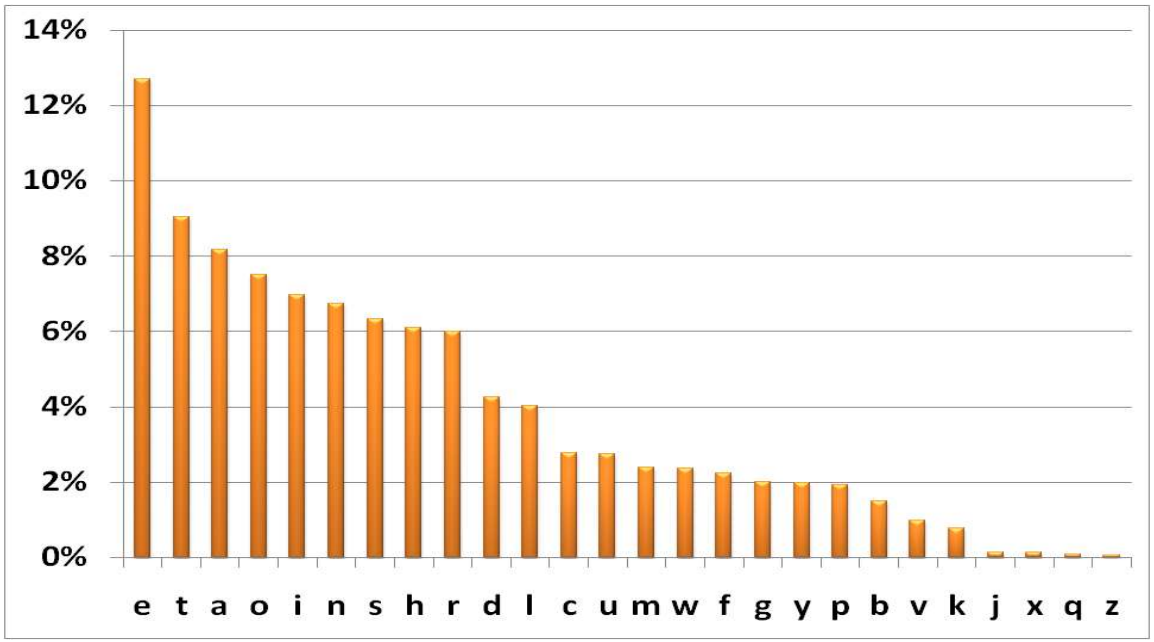
- **An encryption algorithm is computationally safe if ..**
  - Cost of breaking the cipher is much greater than the value of the encrypted information
  - Time to break the cipher is much longer than the useful lifetime of the encrypted information

Breaking Simple Codes

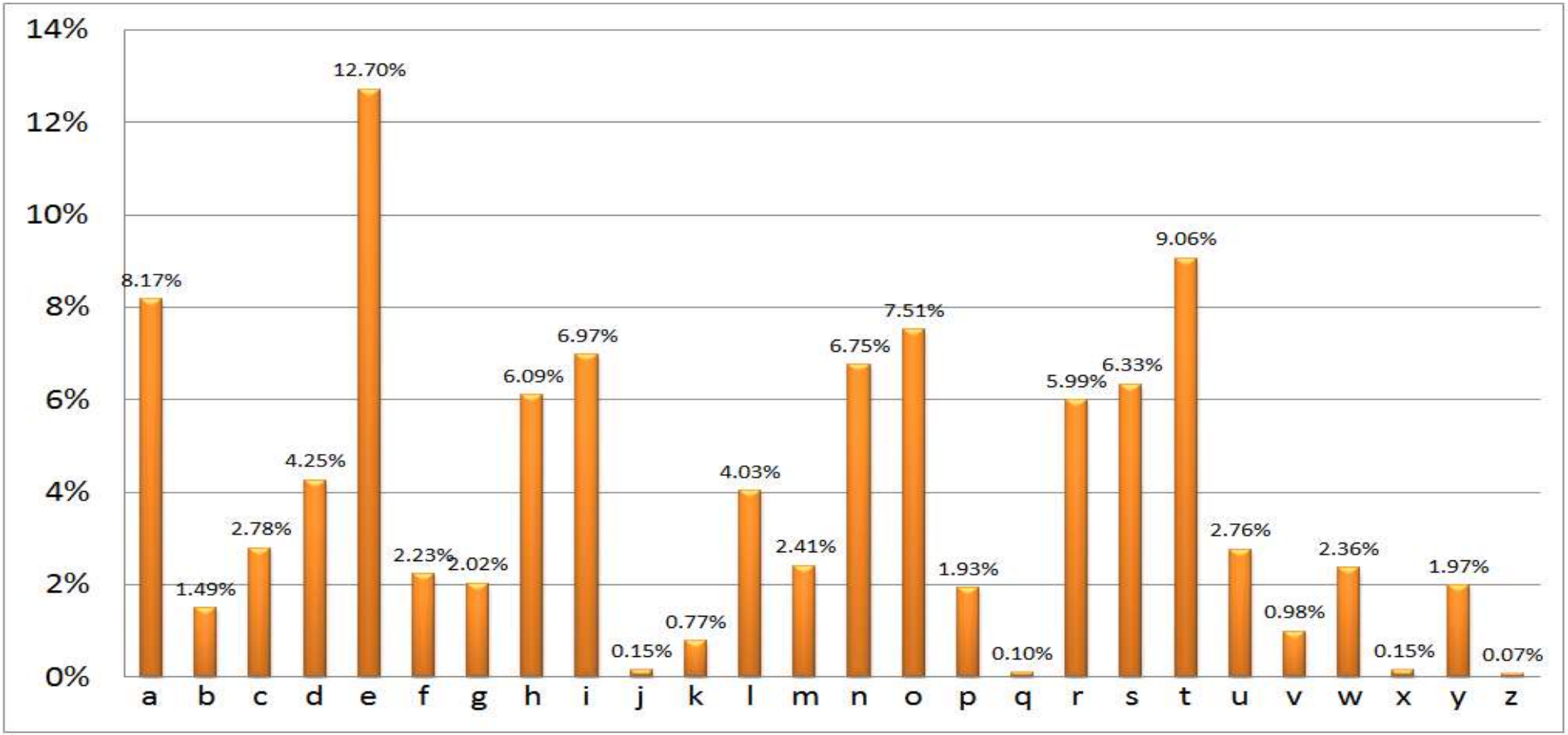
- How do we break Caesar cipher?
- If the plaintext is English text then exploit the regularities of the language

- **Example**

fubswrorjblvwkhvflhqfhr
ivhfuhwzulwlqjrilwvxqdxwk
rulchgghfubswlrqdqgriwk
huxohvzklfkduhlqwxuqlqwhq
ghgwrpdnhwkdwxqdxwkrulc
hgghfubswlrqpruhgliilfxow



Frequency Analysis



Back to Caesar and Vigenère

- **In either method,**
  - How difficult is to implement?
  - How difficult is to crack it using a computer?

Back to Caesar

- **Example:**
  - Original text:

This is an example of how to test Caesar's method. After we take this example, we remove all punctuations and spaces from the original text. The outcome from this process is the 'plaintext' we require.

Back to Caesar

- **Plaintext:**

thisisanexampleofhowtotestcaesarsmethodafterwetaket
hisexampleweremoveallpunctuationsandspacesfromth
eoriginaltexttheoutcomefromthisprocessistheplaintext
werequire

Back to Caesar

- Choose a key: key='**f**'
- Convert the letters to numbers

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- Numerical value of key and ciphertext (key=**5**)
- Convert by adding the key and taking modulo 26 of the result

T	H	I	S	I	S	…
19	7	8	18	8	18	…
24	12	13	23	13	23	…
Y	M	N	X	N	X	…

Searching the Key

- **This is the ciphertext:**

YMNXNXFSJCFRUQJTKMTBYTYJXYHFJXFWXRJYMTIFKYJWBJYFPJ
YMNXJCFRUQJBJWJRTAJFQQUZSHYZFYNTSXFSIXUFHJXKWTRY
MJTWNLNSFQYJCYYMJTZYHTRJKWTRYMNXUWTHJXXNXYMJU
QFNSYJCYBJWJVZNWJ

Frequency Analysis

- Count the occurrences of letters in the ciphertext

A	B	C	D	E	F	G	H	I
1	4	4	0	0	13	0	5	2
J	K	L	M	N	O	P	Q	R
24	4	1	8	10	0	1	6	7
S	T	U	V	W	X	Y	Z	
6	12	6	1	9	13	19	4	

Frequency Analysis

- Divide occurrences / (sum of all the frequencies)

Our example

A	B	C	D	E	F	G	H	I
0.006	0.025	0.025	0.000	0.000	0.081	0.000	0.031	0.012
J	K	L	M	N	O	P	Q	R
0.149	0.025	0.006	0.050	0.062	0.000	0.006	0.037	0.044
S	T	U	V	W	X	Y	Z	
0.037	0.075	0.000	0.006	0.056	0.081	0.118	0.025	

English text frequency

A	B	C	D	E	F	G	H	I
0.082	0.015	0.028	0.043	0.127	0.022	0.020	0.061	0.070
J	K	L	M	N	O	P	Q	R
0.002	0.008	0.040	0.024	0.067	0.075	0.019	0.001	0.060
S	T	U	V	W	X	Y	Z	
0.063	0.091	0.028	0.010	0.024	0.002	0.020	0.001	

The key is a shift of five letters, i.e. the letter 'F'

Using vectors to find the key

- Write English text frequencies as a vector

$$\overline{A}_0 = (0.082, 0.015, 0.028, 0.043, ..., 0.001)$$

If $\overline{A}_0 = (f_0, f_1, f_2, \ldots, f_{25})$

and $\overline{A}_j = (f_j, f_{j+1}, \ldots f_{25}, f_0, \ldots, f_{j-1})$

Where $\overline{A}_j$ represents $\overline{A}_0$ shifted by *j* spaces to the right

Then the dot product is:

$$\overline{A}_i \cdot \overline{A}_j = f_i f_j + f_{i+1} f_{j+1} + f_{i+2} f_{j+2} + \ldots$$

- Examples:

$$\overline{A}_0 \cdot \overline{A}_0 = (0.082)^2 + (0.015)^2 + (0.028)^2 + \ldots + (0.001)^2 = 0.066$$

$$\overline{A}_0 \cdot \overline{A}_1 = 0.082 \times 0.015 + 0.015 \times 0.028 + \ldots + 0.001 = 0.039$$

Using Vectors to find the key

- Properties:

  * **Symmetrical** $\overline{A}_i \cdot \overline{A}_j = \overline{A}_j \cdot \overline{A}_i = 0.066$

  * **The largest value is when *i* = *j***

|i - j|	0	1	2	3	4	5	…
$\mathbf{A_i.A_j}$	0.066	0.039	0.032	0.034	0.044	0.033	…

- Finding the key:

  * **Write the ciphertext frequencies as a vector**

  $$\overline{W} = (0.006, 0.025, 0.025, 0.000, \ldots)$$

  * **Evaluate**

  $$\overline{W} \cdot A_0, \overline{W} \cdot A_1, \overline{W} \cdot A_2, \overline{W} \cdot A_3, \ldots, \overline{W} \cdot A_{25}$$

|i - j|	0	1	2	3	4	5	…
$\mathbf{W.A_j}$	0.028	0.04	0.035	0.029	0.036	**0.066**	…

Breaking the Vigenère cipher

- **How do we break Vigenère code?**

FHYULCVBYEBYJEUDSYQEAFELWRGFGCQI
SVBCVTIQOUQFMUDCYEJRPGQGRKEZOUCS
RGQTDRRRKEKRDCUNARMNXTCUHCZAQWHC
VOLRFZHNHDMGQBYEBYJEYZEYOTFBLMQD
MQBYQKCUHCDPNOICGHGVGCQISVTMPALB
PPRBJHMQKIQLNTHNRLOLVILFLSGERKEQ
SECGOKHTCUALGTFHCMZCYWCFHRRKEJHT
RHRGVEJHTRHRCHECH

Guessing the key length..

Vigenère cipher broken
by Charles Babbage

FHYULCVBYEBYJEUDSYQEAFELWRGFGCQI
SVBCVTIQOUQFMUDCYEJRPGQGRKEZOUCS
RGQTDRRRKEKRDCUNARMNXTCUHCZAQWHC
VOLRFZHNHDMGQBYEBYJEYZEYOTFBLMQD
MQBYQKCUHCDPNOICGHGVGCQISVTMPALB
PPRBJHMQKIQLNTHNRLOLVILFLSGERKEQ
SECGOKHTCUALGTFHCMZCYWCFHRRKEJHT
RHRGVEJHTRHRCHECH

- Look at the repetitions in the cipher
- How these repetitions relate to the key size?
- The repetitions are multiples of 3. So take every third letter and make frequency analysis.

  FHY**U**LC**V**BY **E**BY**J**EU...
- Can you find the plain text?

Vigenère: Length of key

- Length of the key
- Note that the table shows only part of the plaintext given on slide 77

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

Vigenère: Length of key

- Copy and shift the ciphertext by one..
- Look for coincidences

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

Vigenère: Length of key

- Copy and shift the ciphertext by two..
- Look for coincidences

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

Vigenère: Length of key

- Copy and shift the ciphertext by three..
- Look for coincidences

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

Vigenère: Length of key

- For the whole ciphertext:

Displacement	1	2	3	4	5	6	7	8	9	10
Coincidences	4	12	25	8	6	25	8	8	27	5

- If the displacement is a multiple of three we have a large number of coincidences

- Most probably the key is of size 3

Vigenère: finding the key

- **Original ciphertext:**
  FHYULCVBYEBYJEUDSYQEAFELWRGFGCQI…
- **Split the ciphertext in three list**
- **First list contains, 1st, 4th, 7th .. Letters**
  FUVBESEERG…
- **Second list contains, 2nd, 5th,8th.. Letters**
  HLBYUYALGC…
- **Third list contains, 3rd,6th,9th.. Letters**
  YCEJDQFWFQ…
- **Now do frequency analysis on each of the list letters**

Vigenère: finding the key

- **If the key is of size n then**
  - For *i*=1 .. n
    1. Compute the frequencies of the letters in positions *i* mod *n* and make the vector ***W***
    2. For *j* = 0 ... 25 compute p_j = $\boldsymbol{W.A}_j$
    3. Let k_i = *j* where p_j is the maximum value
  - The key is probably $\{k_1,k_2,...,k_n\}$
- **Our example**
  - Key = {3,0,24} = DAY

Vigenère: Plaintext

CHARLESBABBAGEWASANECCENTRICGEN

IUSBESTKNOWNFORDEVELOPINGTHEBLU

EPRINTFORTHEMODERNCOMPUTERHEWAS

THESONOFBENJAMINBABBAGEAWEALTHY

LONDONBANKERHEAPPLIEDHISGENIUST

OMANYPROBLEMSHISINVENTIONSINCLU

DETHESPEEDOMETERANDTHECOWCATCHE

RTHELETTERISELETTEREEEEE

Vigenère: Plaintext

CHARLES BABBAGE WAS AN ECCENTRIC GEN
IUS BEST KNOWN FOR DEVELOPING THE BLU
EPRINT FOR THE MODERN COMPUTER HE WAS
THE SON OF BENJAMIN BABBAGE A WEALTHY
LONDON BANKER HE APPLIED HIS GENIUS T
O MANY PROBLEMS HIS INVENTIONS INCLU
DE THE SPEEDOMETER AND THE COW CATCHE
R THE LETTER IS E LETTER EEEEE

Vigenère Cipher: Cryptanalysis

1. Determine the key length of the keyword (m)

- **Kasiski test**:
  - Search ciphertext for pairs of identical segments (Such as the BYs shown on slide 78).

- **Index of coincidence**:
  - Suppose $x=x_1,x_2,\ldots,x_n$ is a string of length n. The index of coincidence of x is defined as the probability that two random elements of x are identical.

2. Determine each of the keys (K_i) separately; hence, $K=(K_1,K_2,..,K_m)$

Applications

Applications of encryption codes

Technology	Comments
WEP 128Bit	Wired Equivalent Privacy, the security system built into 802.11b wireless LAN equipment. Its RC4 basses encryption was broken by AT&T Engineers.
CMEA 64bit	Cellular Message Encryption Algorithm, this is supposed to ensure privacy on digital cell phones.
DES 56bit	Digital Encryption Standard, used throughout the Internet and other systems.
PGP 56bit	Pretty Good Privacy, a popular e-mail and file security program.
S/MIME 40bit	An RSA based encryption system to earlier versions of secure Outlook Express, and some other e-mail systems
CLSID Microsoft MSN and e-mail security. Microsoft SSH 40bit	SSH is a widely used client-server application for authentication and encryption of network communications.
QNX	Stock Exchange's facility security system, and VISA International's transaction processing and verification system.

Applications of encryption codes

Technology	Comments
RSA RC5 56bit	RC5 Is one of the more common implementations by RSA. It is used widely throughout business and the Internet.
SDMI	Secure Digital Music Initiative is the digital watermarking system designed to prevent MP3's from being copied.
RC4/MD5 128bit	This is the security used in all of Microsoft's "Office" products for password security, core design by RSA.
SSL/RC4 128bit	The Secure Socket Layer or SSL is based on RSA's RC4 and has been hacked in its "strong" form. This is the "secure" in virtually every online credit card ordering system and secure web page.
GSM Phones	The algorithm that secures more that GSM digital phones worldwide.

Summary

- **Attacks, services and mechanisms**

- **Introduction to:**
  - Encryption as a security mechanism
    - Classification of encryption mechanisms
    - Substitution-Transposition, Block-Stream, Symmetric-Asymmetric
  - Key agreement

- **Examples of breaking simple codes**

Conventional Encryption

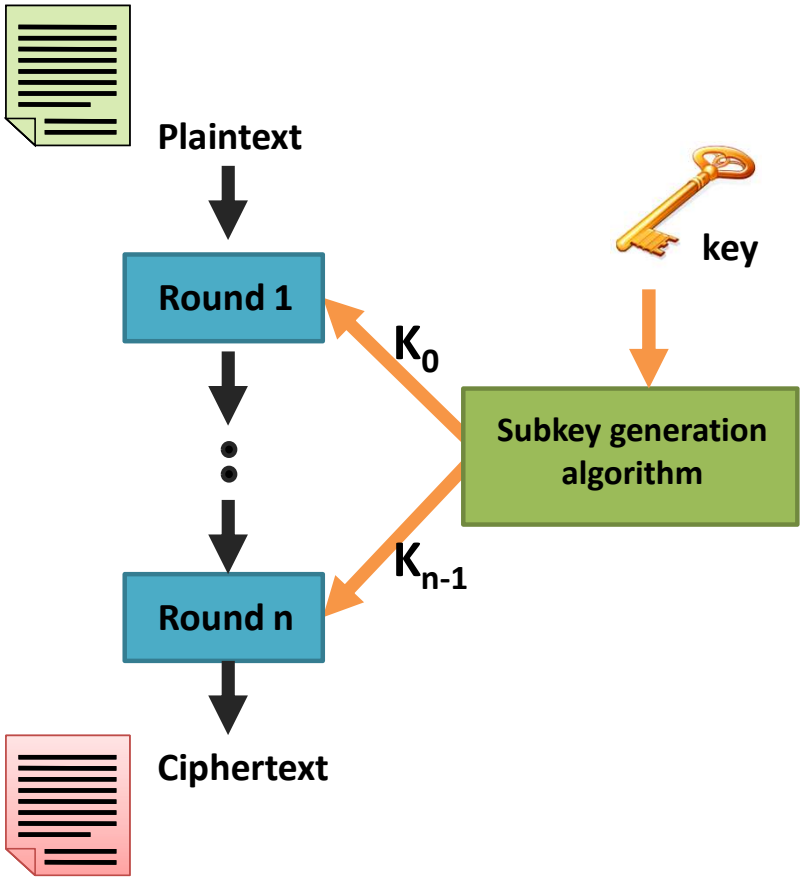
Block Ciphers

- A block cipher process one block of elements of the plaintext at a time.
- It produces one block of ciphertext of same size as the plaintext.
- Block ciphers that use a shared key (symmetric) are known as **conventional encryption** algorithms.

Horst Feistel of IBM in 1973 [FEIS73]

Feistel Algorithm: Encryption

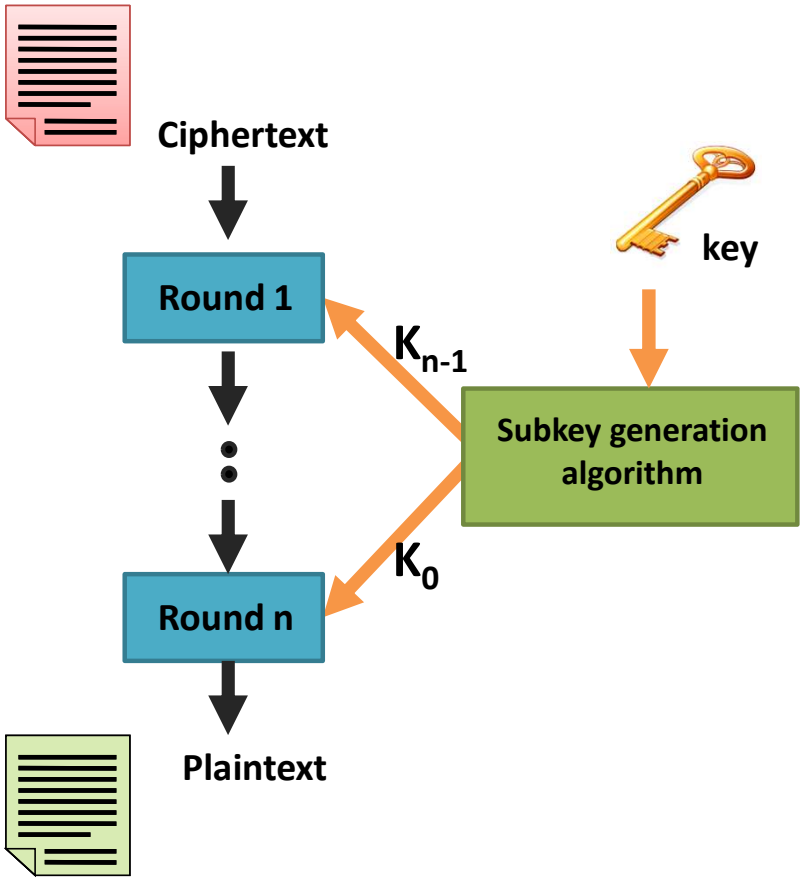
- **Design Features**
  1. Block size
  2. Key size
  3. Number of rounds
  4. Sub-key generation algorithm
  5. Round Function



Shannon's "Confusion and Diffusion"

Feistel Algorithm: Decryption

- **Design Features**
  1. Block size
  2. Key size
  3. Number of rounds
  4. Sub-key generation algorithm
  5. Round Function



Data **E**ncryption **S**tandard - **DES**

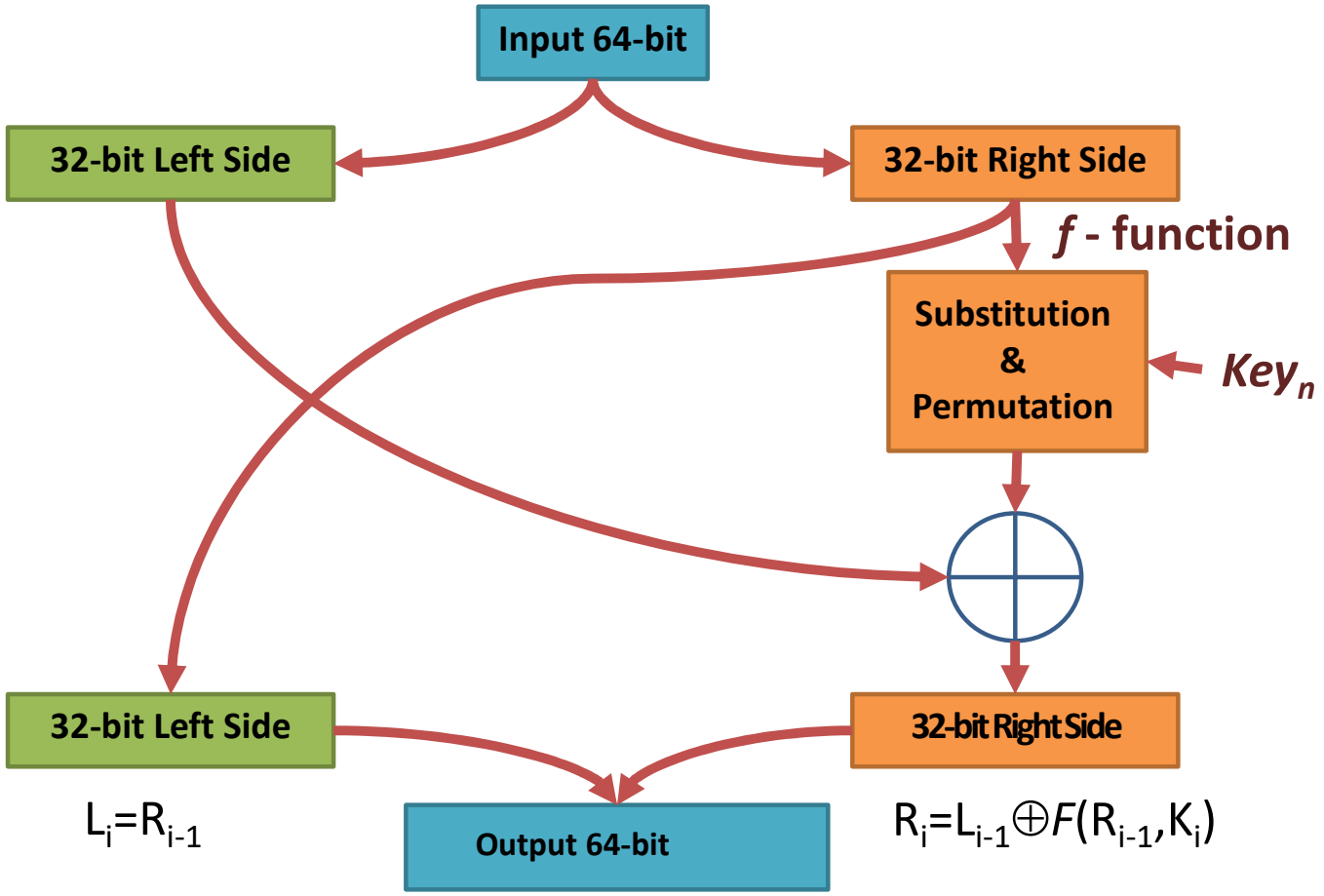
- Adopted by the National Bureau of Standards
  // National Institute of Standards and Technology (NIST) // in 1977.

- Is a block cipher with:
  - **Block size :** 64 bits
  - **Key size :** 56 bits
  - **Number of rounds:** 16

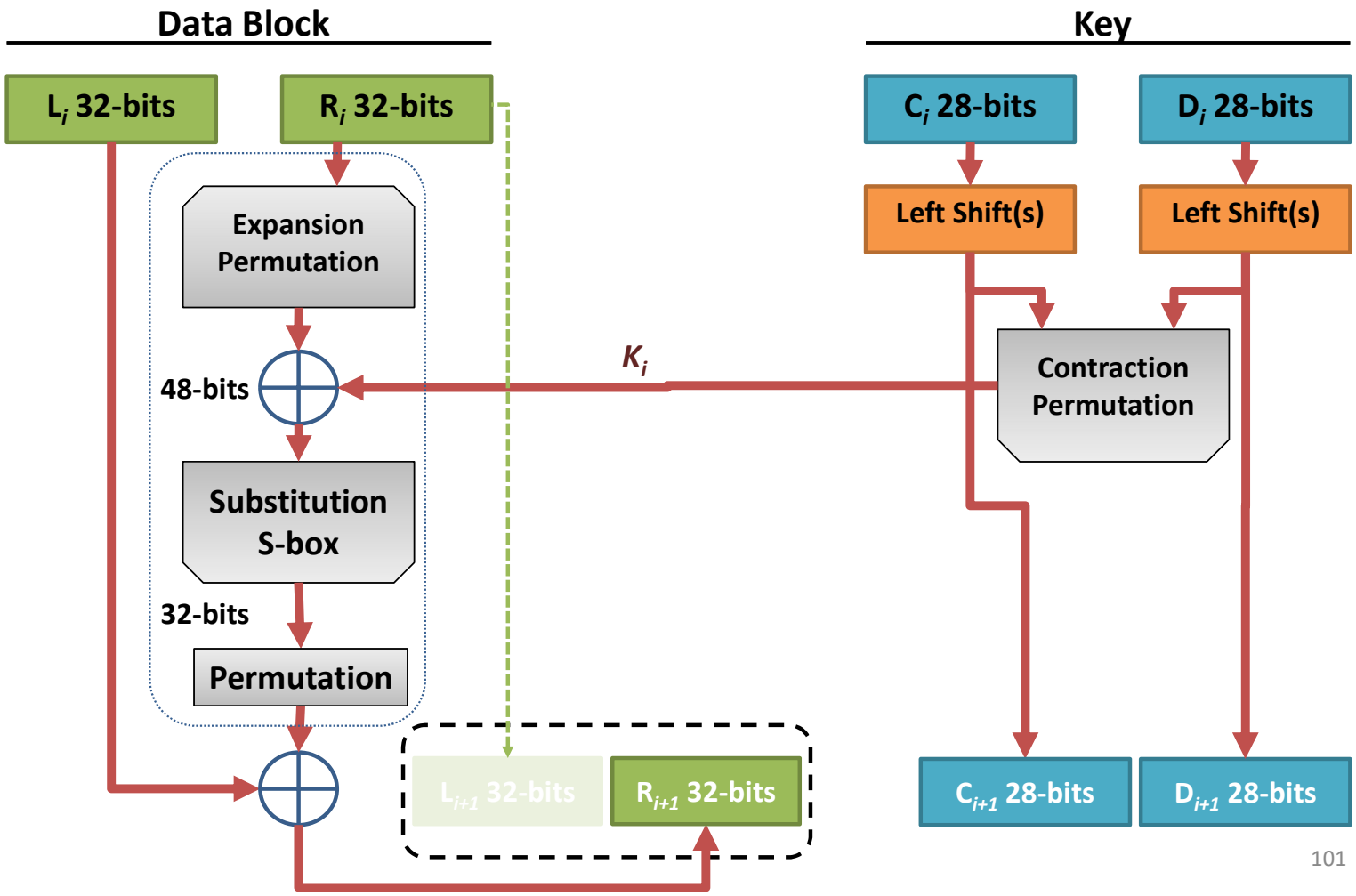
- First the plaintext passes through an initial permutation (T)

- The plaintext (ciphertext) is divided in two parts (Left and Right)

One Round in DES

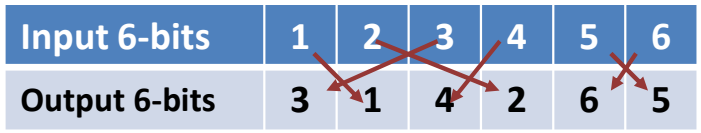


DES: Substitution and Transposition

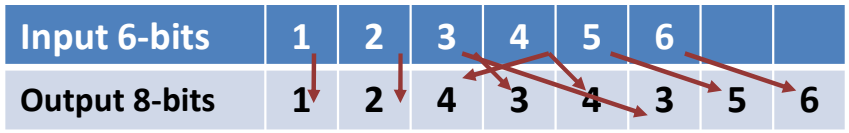


Permutations (transposition)

- Example with a 6-bit block. Note this is not how it is done in DES as the block sizes are different.



Permutation



Expansion Permutation

- **Notation:**
  - Permutation (3,1,4,2,6,5)
  - Expansion Permutation (1,2,4,3,4,3,5,6)

- In DES the expansion permutation is from 32-bits to 48-bits.

S-box (Substitution)

- Take 6-bits block $b_0\ b_1\ b_2\ b_3\ b_4\ b_5$
- Take the first and last bit, $b_0\ b_5$, this represents a binary number (from 0 to 3 in decimal), let call this number row.
- Take the rest of the bits, $b_1\ b_2\ b_3\ b_4$, this represents a binary number (from 0 to 15), call this number the column.
- Use the row and column value to read the number in the S-box. This number is the output of the S-box, the substitution

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	15	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	2	14	10	0	6	13

Notice that we put the row and columns and the table entries using decimal numbers (instead of binary)

Example: S-box

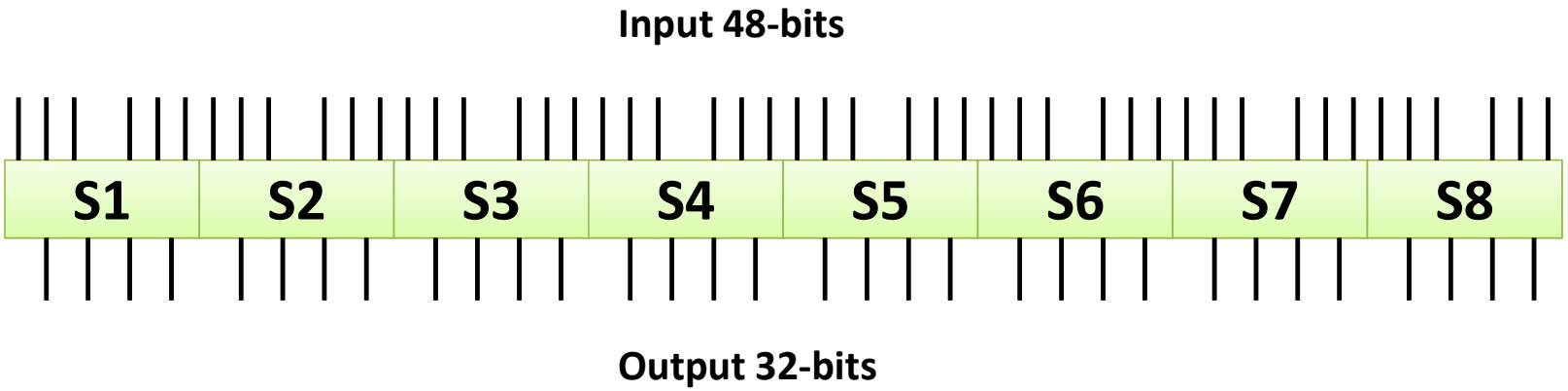
- Suppose that the binary number is 010110
- The first and last bits are 00, in decimal = 0
- The rest of the bits are 1011, in decimal = 11
- Use the table to find row 0, column 11

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	15	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	2	14	10	0	6	13

- The entry value is 12 which in binary is 1100.
- The output of the S–box is 1100

S-box

- DES uses 8 S-boxes for the substitutions



Key-generation

- The algorithm expects a 64-bit long key
- Every 8-bits of this key is ignored (giving 56-bit key)
- The key bits are subjected to a permutation
- The 56-bit key is split into two parts C_i and D_i ; each of 28-bits long.
- At each round C_i and D_i are subject to a circular left shift

$$b_{27}b_{26}...b_1b_0 \leftarrow b_0b_{27}...b_2b_1$$

(The number of bits shifted depends on the round)

- The sub-key is submitted to a contraction permutation (48-bit output)

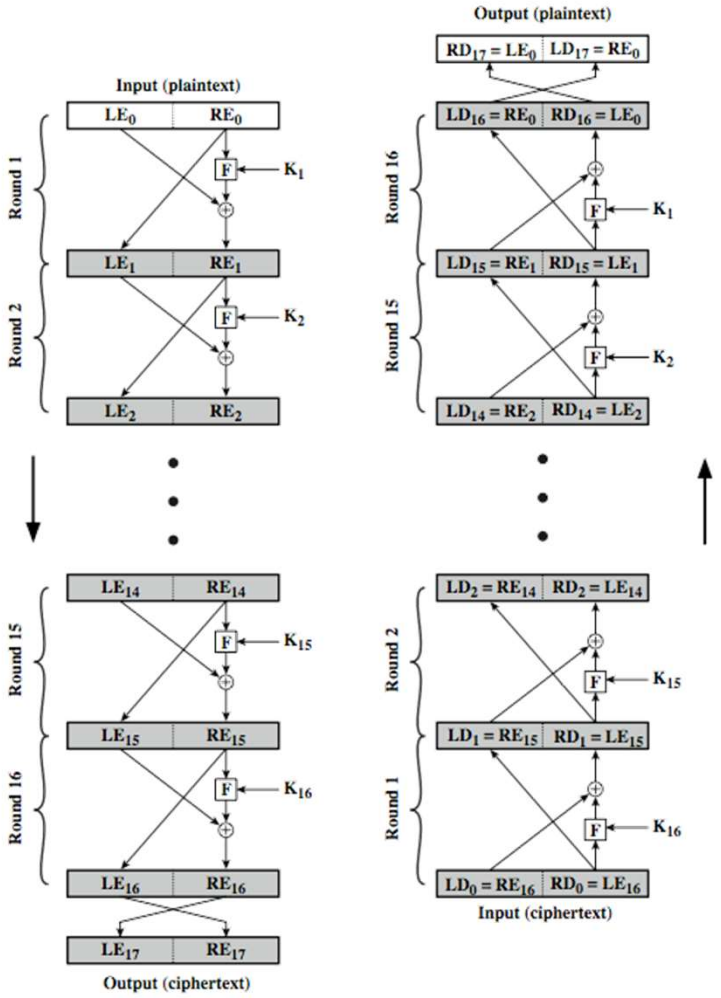
DES: notation

Process per iteration is

$h_i : (R_i, L_i) \rightarrow (R_i \ , \ L_{j-1} \oplus f(R_i, K_i))$

and the swapping

$g : (R_i, L_i) \rightarrow (L_i, R_i)$



Permutation (T) and inverse permutation (T^{-1})

The strength of DES

- **There are two concerns:**
  - Cryptanalysis by exploiting the characteristics of the algorithm. However, there are relatively few weaknesses in the algorithm.
  - The key length:

Key size in bits	Number of different keys	Time required at 10^6 Decryptions/μs
32	2^{32}	2^{31}μs=2.15ms
56	2^{56}	2^{55}μs=10hrs
128	2^{128}	2^{127}μs=5.4X10^6 yrs
168	2^{168}	2^{167}μs=5.9X10^{30} yrs

DES

- DES with 56-bit key, **Don’t use it, not secure enough**
- DES with 128-bit key, **secure now?**

Double DES

- Encrypt the same plaintext multiple times using DES with different keys.
- If simple DES is using a key of 56-bits then the keyspace consists of 2^{56} keys.
- Notation:
  - $E_k(m)$ is the encryption function E with key K and m is the message.
- Double DES will be $E_{K1}(E_{K2}(m))$, where K_1 and K_2 are the keys.
- If the keys are of length 56-bits then it seems that in double DES the key space consist of 2^{112} keys.

Is that so?

- However, this is not true, double DES has the security level of a 57-bit key.

Meet-in-the-middle attack

- Alice and Bob are going to use double DES
- They know the keys K_1 and K_2.
- Notation:
  - *E* means encryption and *D* means decryption.
- Alice sends to Bob the encrypted message $c = E_{K1}(E_{K2}(m))$.
- Bob decrypts the message $m=D_{K2}(D_{K1}(c))$.
- Alice and Bob believe that Eve (the hacker) will need to discover both keys K_1 and K_2 by brute force to decrypt the message.

Meet-in-the-middle attack

- Eve has intercepted the message m and $c = E_{K1}(E_{K2}(m))$.
- She wants to find K_1 and K_2 .
- She computes $E_K(m)$ for all possible keys and stores the results in a list.
- She computes $D_K(c)$ for all possible keys and stores the results in a list
- She compares the two lists, and looks for a match
- If she found a match, then Eve knows K_1 and K_2.

Triple DEA (TDEA)

- **TDEA uses three keys executions of the DES algorithm**

  c = $DES_{K3}(DES_{K2}^{-1}(DES_{K1}(m)))$

  where *c* = ciphertext, *m* = plaintext

- **Notation:**
  - $DES_K(X)$ = encryption of X using key *K*
  - $DES_K^{-1}(X)$ = decryption of X using key *K*

TDEA

- **Decryption is achieved using**

$$m = DES_{K1}^{-1}(DES_{K2}(DES_{K3}^{-1}(c)))$$

- **Key length 168-bits long**

The Advanced Encryption Standard

- The National Institute for Security Technologies (NIST) – USA, after 4 years of consideration has introduced Advanced Encryption Standard (AES) to replace the previous DES.
- Introduced in November 2001, the standard uses the Rijndael algorithm.
- Developed by Joan Daemen and Vincent Rijmen (Belgium).

The Advanced Encryption Standard

- Rijndael is an iterated block cipher. Each intermediate cipher result is called a '**state**'.
- Rijndael can operate over a variable-length block using variable-length keys; (128-,192-,256- bit).
- AES only supports a 128-bit block size.
- The algorithm is written so that block length and/or key length can easily be extended in multiples of 32 bits.
- Does not use a Feistel structure as it process the entire data block in parallel during each round using substitutions and linear transformations.
  - In the classic Feistel structure, half of the data block is used to modify the other half of the data block, and then the halves are swapped.

The cipher Rijndael

Example: 128bit Key (10 rounds)

- An initial round - key addition;
- N_{r-1} rounds;
- A final round.

Characteristics
- Immune from all known attacks.
- Fast/compact on various platforms
- Design simplicity.

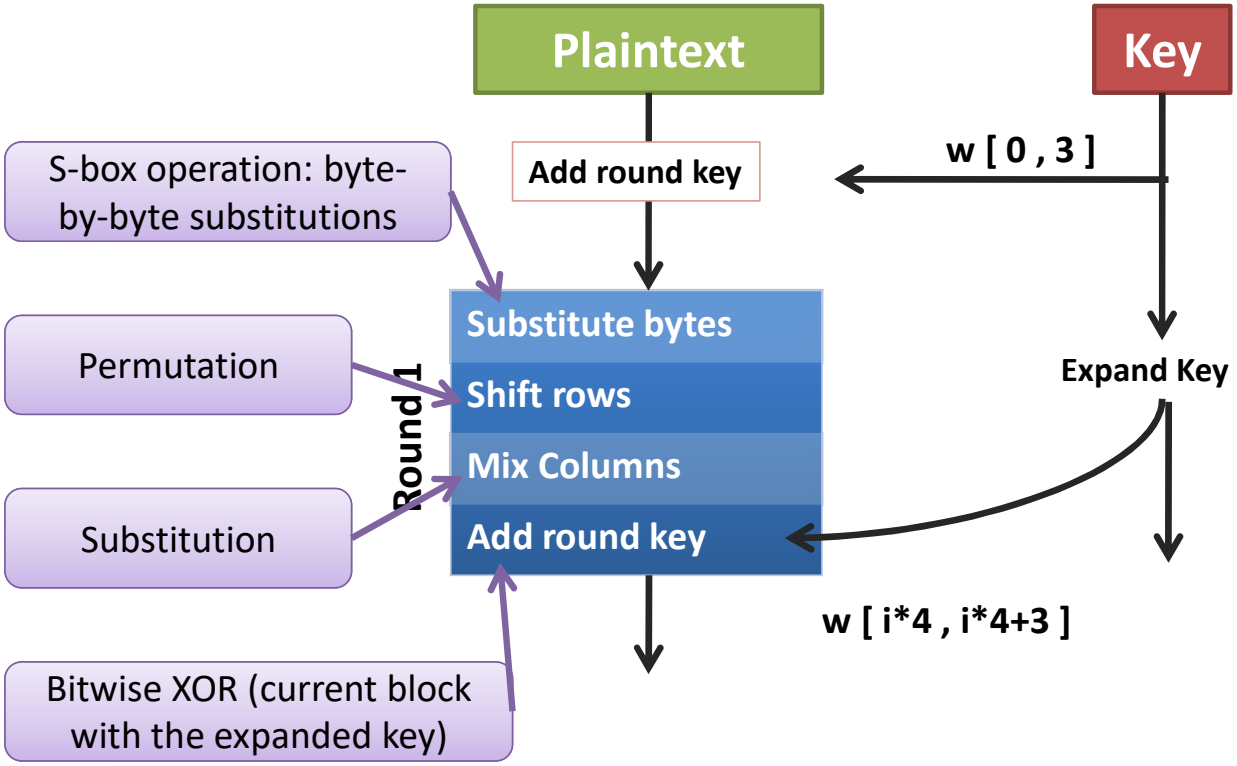
- In pseudo code (taken from the revised AES proposal)

```
Rijndael(State,CipherKey)
  {
  KeyExpansion(CipherKey,ExpandedKey);
  AddRoundKey(State,ExpandedKey);
  For(i=1; i<Nr; i++)
  Round(State,ExpandedKey + Nb*i);
  FinalRound(State,ExpandedKey + Nb*Nr);
  }
```

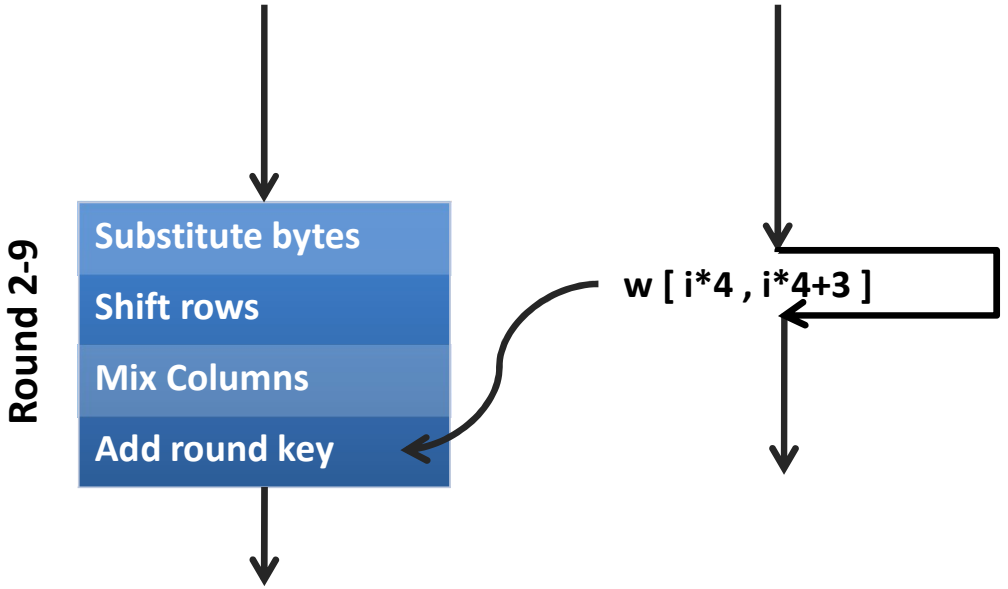
Key Size (Bits)	128	192	256
Block size (Bits)	128	128	128
Number of rounds	10	12	14
Round Key Size (Bits)	128	128	128
Expanded Key Size (Bytes)	176	208	240

Typical AES parameters

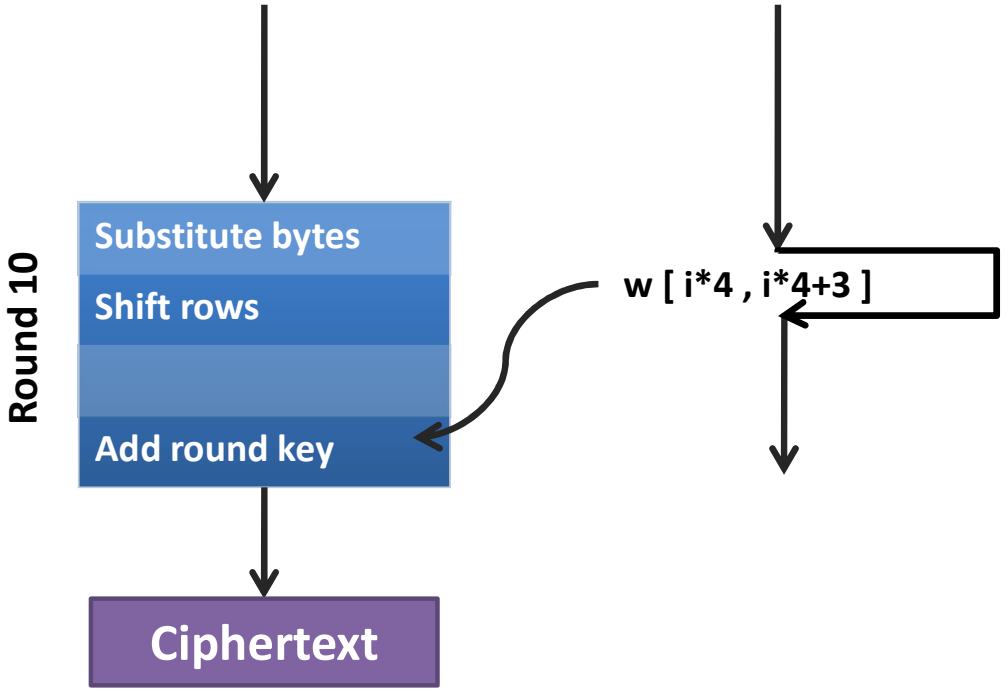
The cipher Rijndael



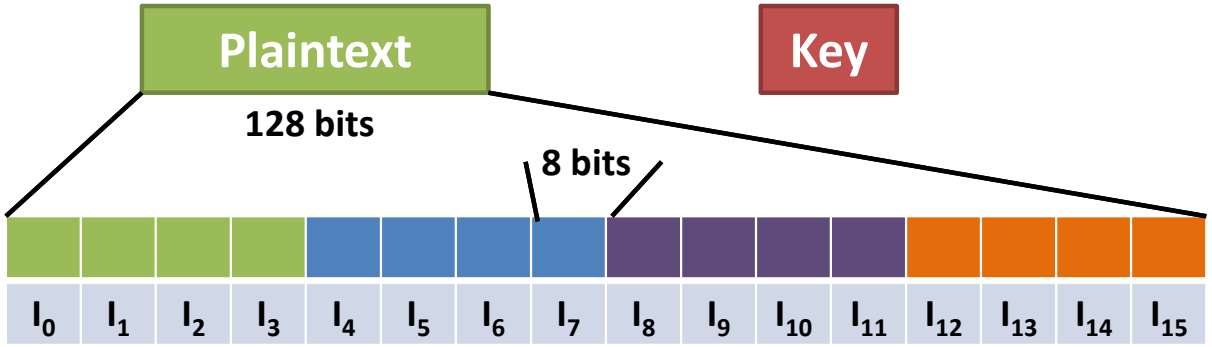
The cipher Rijndael



The cipher Rijndael



The cipher Rijndael



Divide the plaintext in 16 blocks

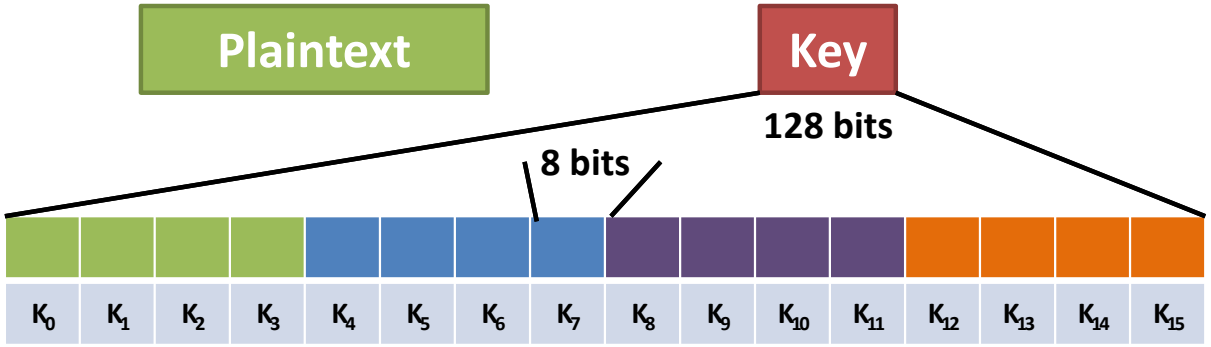
With the blocks form a 4 X 4 matrix

I_0	I_4	I_8	I_{12}
I_1	I_5	I_9	I_{13}
I_2	I_6	I_{10}	I_{14}
I_3	I_7	I_{11}	I_{15}

$=$

$$\begin{pmatrix} I_0 & I_4 & I_8 & I_{12} \\ I_1 & I_5 & I_9 & I_{13} \\ I_2 & I_6 & I_{10} & I_{14} \\ I_3 & I_7 & I_{11} & I_{15} \end{pmatrix}$$

The cipher Rijndael



Divide the plaintext in 16 blocks

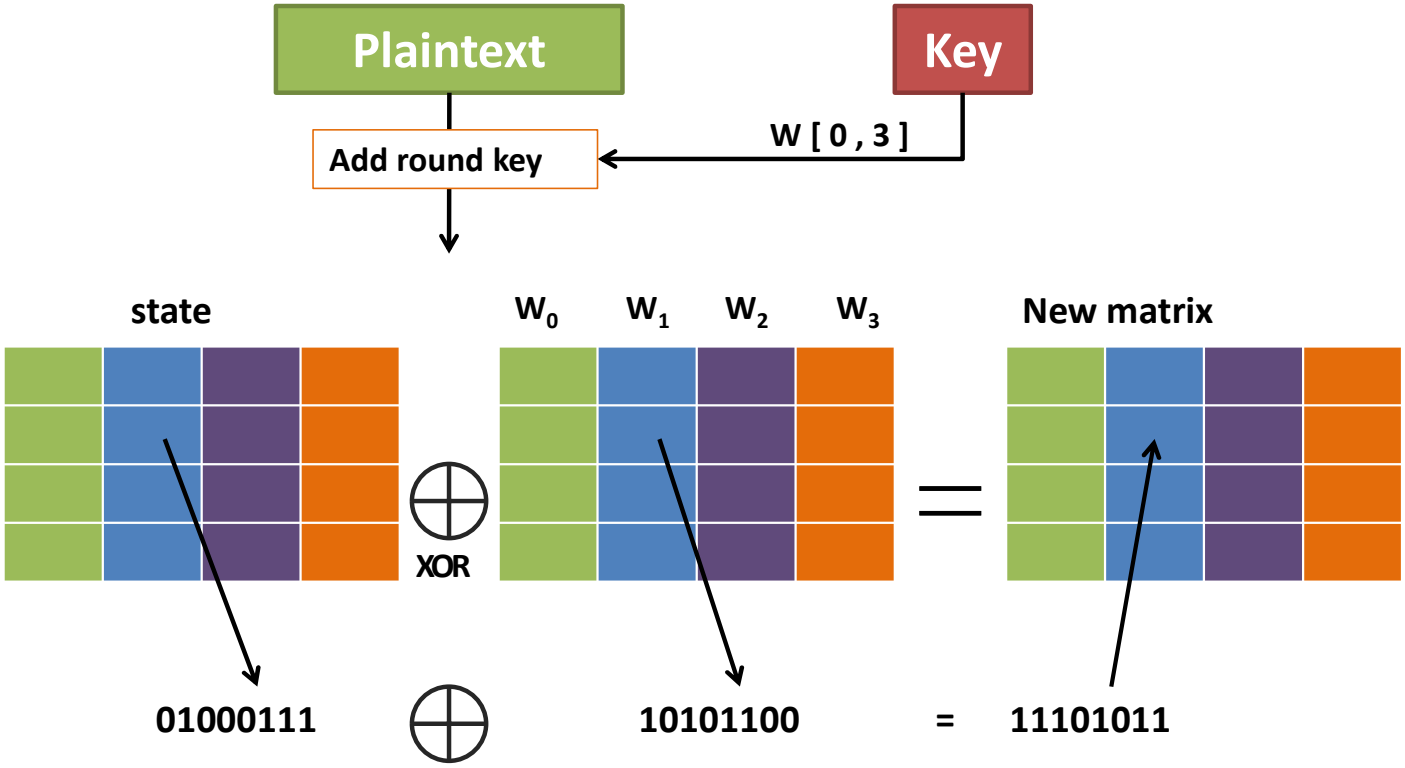
With the blocks form a 4 X 4 matrix

W_0	W_1	W_2	W_3
K_0	K_4	K_8	K_{12}
K_1	K_5	K_9	K_{13}
K_2	K_6	K_{10}	K_{14}
K_3	K_7	K_{11}	K_{15}

=

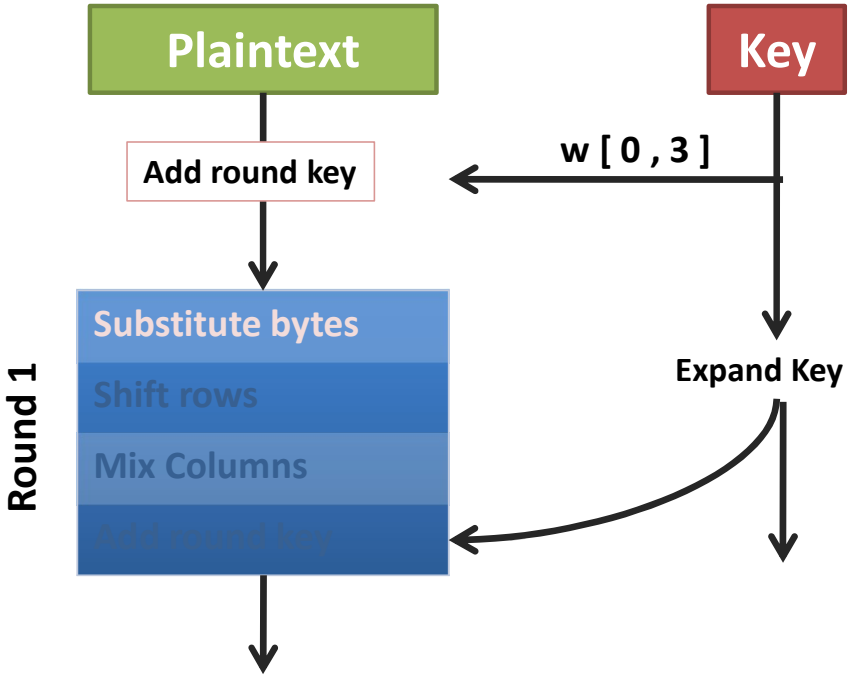
$$\begin{array}{c|c|c|c} W_0 & W_1 & W_2 & W_3 \\ \hline K_0 & K_4 & K_8 & K_{12} \\ K_1 & K_5 & K_9 & K_{13} \\ K_2 & K_6 & K_{10} & K_{14} \\ K_3 & K_7 & K_{11} & K_{15} \end{array}$$

The cipher Rijndael



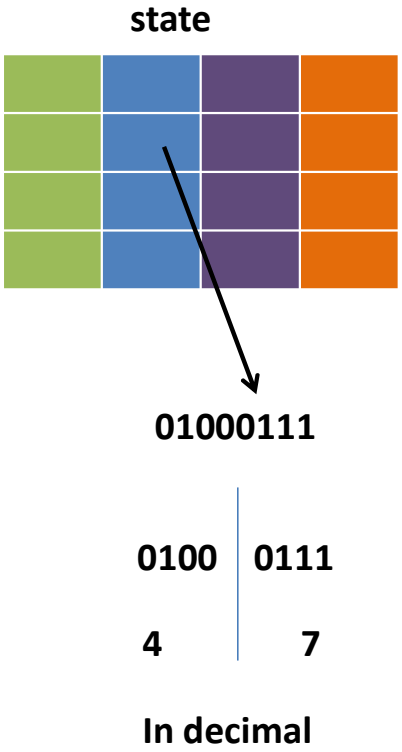
In hexadecimal 47 ⊕ AC = EB

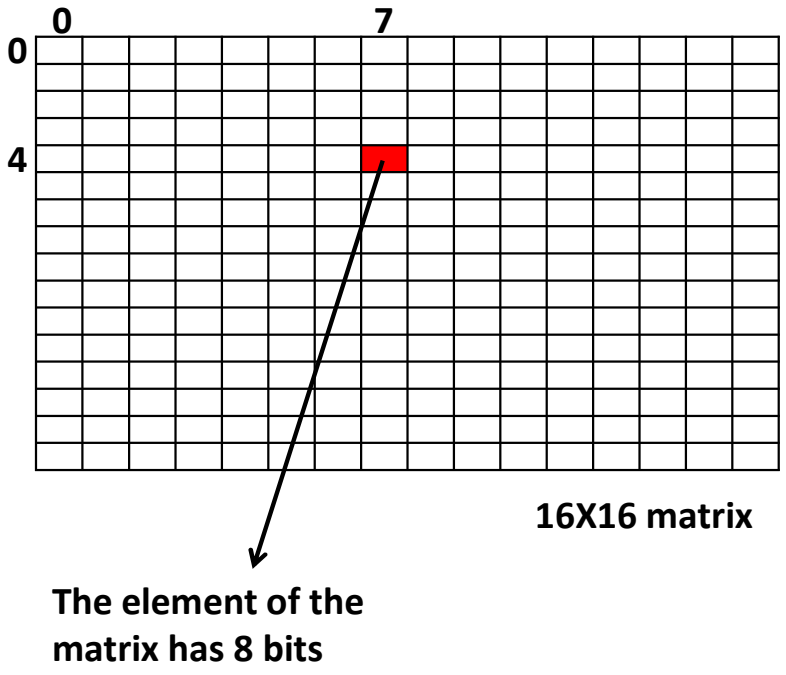
The cipher Rijndael



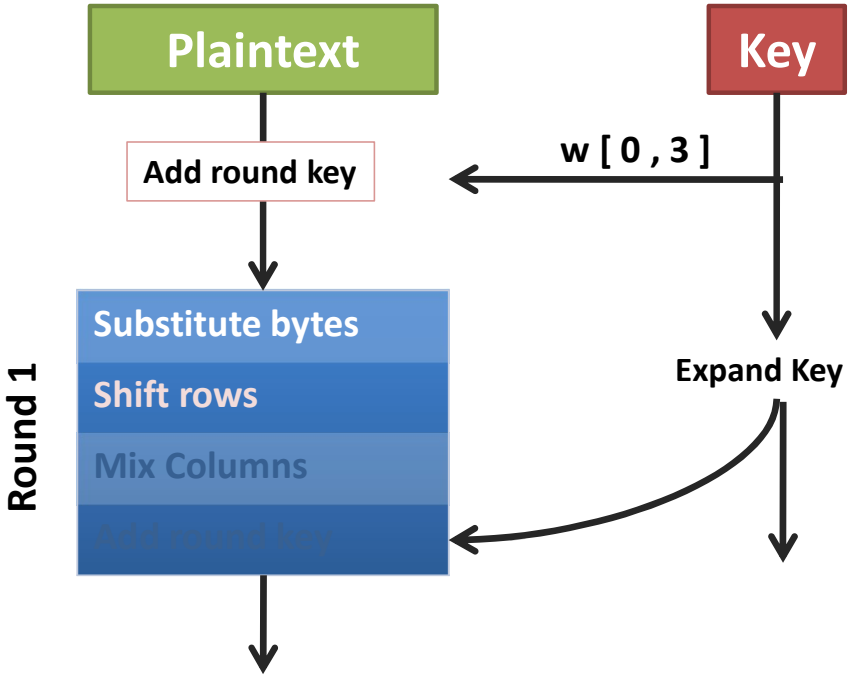
The cipher Rijndael

Substitute bytes



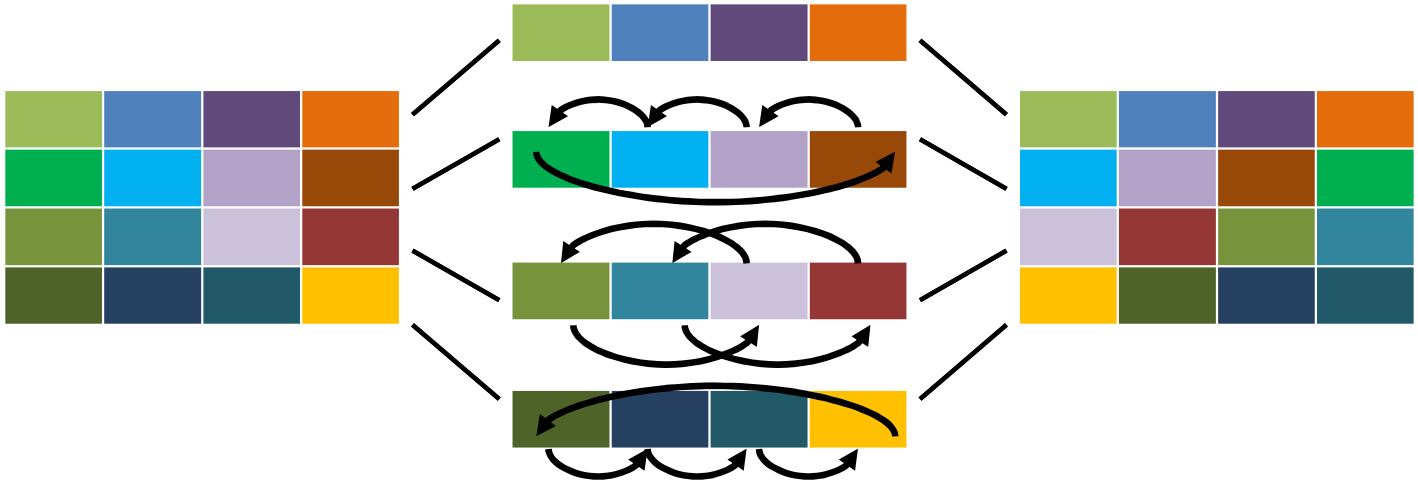


The cipher Rijndael



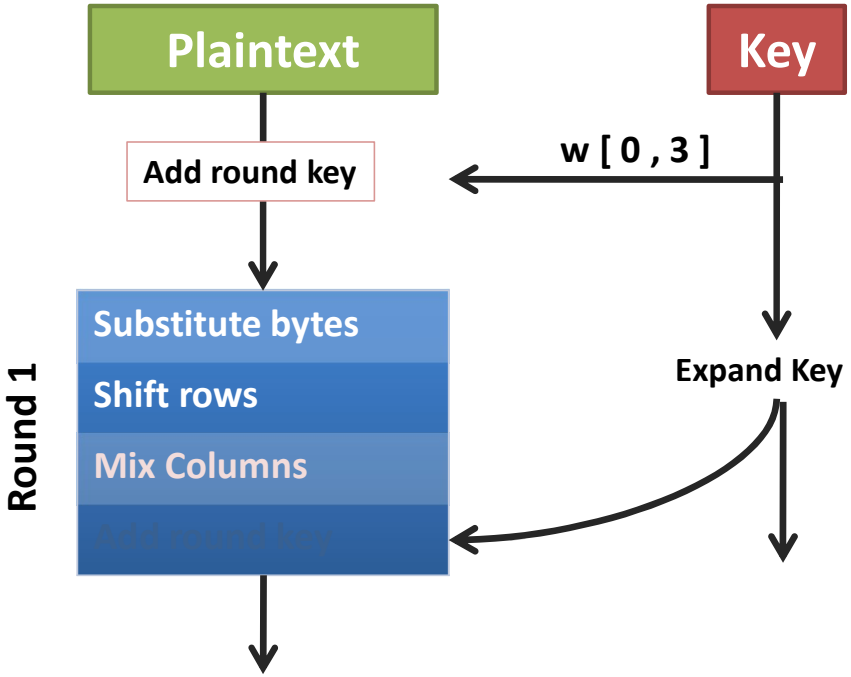
The cipher Rijndael

- Shift rows

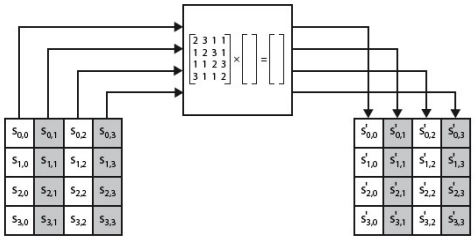


First row stays the same
Second row, 1-byte circular left shift
third row, 2-byte circular left shift
Forth row, 3-byte circular left shift

The cipher Rijndael



The cipher Rijndael



- The mixing of columns is obtained using a matrix multiplication (in the field $\underline{GF(2^8)}$ *GF = Galois* field)

$$\begin{pmatrix} n_{0,0} & n_{0,1} & n_{0,2} & n_{0,3} \\ n_{1,0} & n_{1,1} & n_{1,2} & n_{1,3} \\ n_{2,0} & n_{2,1} & n_{2,2} & n_{2,3} \\ n_{3,0} & n_{3,1} & n_{2,3} & n_{3,3} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{2,3} & S_{3,3} \end{pmatrix}$$

Where 01, 02 and 03 are in hexadecimal (in binary are 01, 10 and 11 respectively) and n_i denotes the new 'State'.

The cipher Rijndael

- **The multiplication is obtained by the sum of multiplying one column by one row (in the field *GF(2^8)*)**

- **Example:**

$$n_{0,0} = (2 \bullet S_{0,0}) \oplus (3 \bullet S_{1,0}) \oplus S_{2,0} \oplus S_{3,0}$$

- **The multiplication using • is as follows:**

If the $S_{i,j} = (b_7,b_6,b_5,b_4,b_3,b_2,b_1,b_0)$ then

$$(2 \bullet S_{i,j}) = \begin{cases} (b_6,b_5,b_4,b_3,b_2,b_1,b_0,0) & \text{if } b_7 = 0 \\ (b_6,b_5,b_4,b_3,b_2,b_1,b_0,0) \oplus (00011011) & \text{if } b_7 = 1 \end{cases}$$

and $(3 \bullet S_{i,j}) = S_{i,j} \oplus (2 \bullet S_{i,j})$

See next slide

Reference to:

- Multiplication in the field $GF(2^n)$ - *No simple straight forward method*
- Multiplication in the field $GF(2^8)$ - *Simple; used in AES*

Multiplication in the field GF(2^8)

- **Consider the finite field used in AES:**

  $m(x) = x^8+x^4+x^3+x+1$

- **Based on the generalised GF(2^n):**

  $x^8 \bmod m(x) = [\, m(x) - x^8 \,] = x^4+x^3+x+1$

- **Consider a polynomial GF(2^8):**

  $f(x) = b_7x^7+ b_6x^6+ b_5x^5+ b_4x^4+ b_3x^3+ b_2x^2+ b_1x+b_0$

  If multiplied by x, then $x*f(x) = (b_7x^8+ b_6x^7+ b_5x^6+ b_4x^5+ b_3x^4+ b_2x^3+ b_1x^2+b_0x) \bmod m(x)$

- **If b7 = 0, then the result is a polynomial of degree less than 8.**

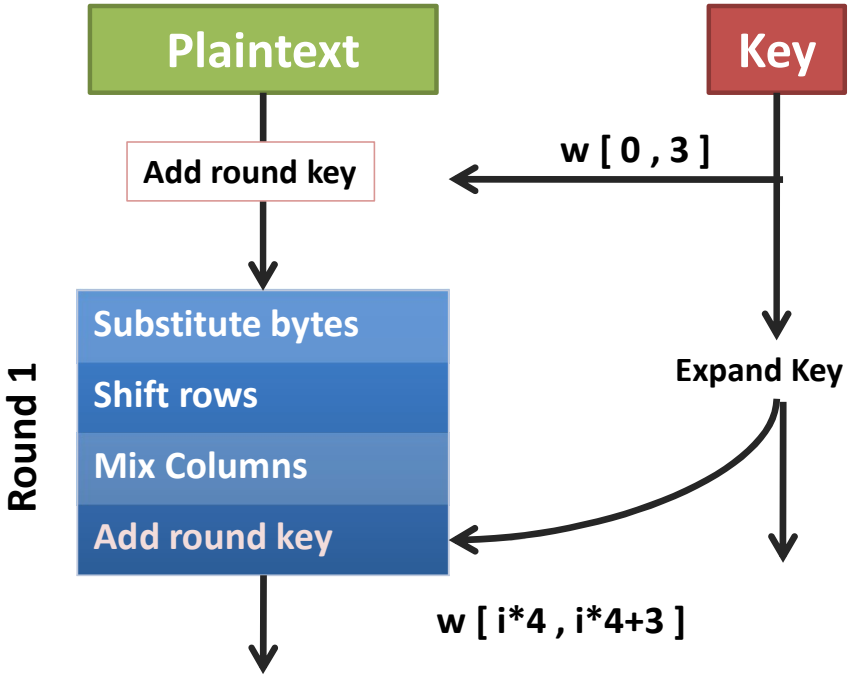
  $b_6x^7+ b_5x^6+ b_4x^5+ b_3x^4+ b_2x^3+ b_1x^2+b_0x$

- **If b7 = 1, then then the reduction modulo *m(x)* is achieved as above.**

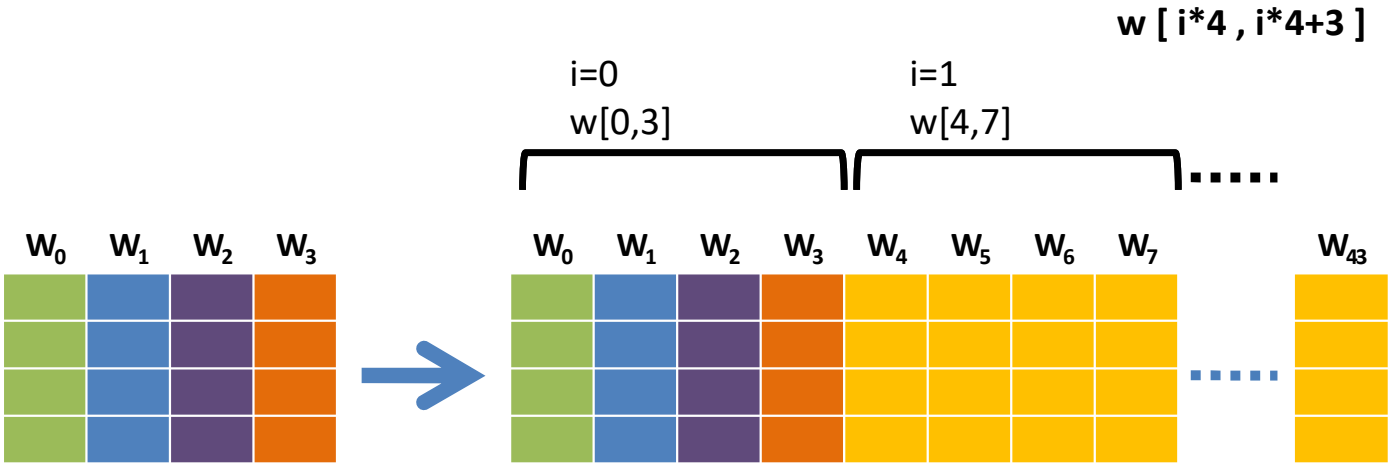
  $b_6x^7+ b_5x^6+ b_4x^5+ b_3x^4+ b_2x^3+ b_1x^2+b_0x + (x^4+x^3+x+1)$

- **Hence, it is implemented as a 1-bit left shift followed by a conditional bitwise XOR with (00011011), which corresponds to (x^4+x^3+x+1).**

The cipher Rijndael



The cipher Rijndael



- This slide explains the algorithm used to expand the key.

The cipher Rijndael: Key expansion

- SubWord = Byte Substitution using the S box
- RotWord = One-byte circular left shift
- Rcon = 'round' constant (given $r(i) = 00000010^{(i-4)/4}$)

```
KeyExpansion(byte key[16], word[44])
  { word tmp;
  for(i=0;i<4;i++)
   w[i]=(key[4*i],key[4*i+1],key[4*i+2],key[4*i+3]);
  for(i=4;i<44;i++){
   tmp=w[i-1];
   if(i%4==0) tmp=SubWord(RotWord(tmp)) XOR Rcon[i/4];
   w[i]=w[i-4] XOR tmp;
  }
  }
```

The cipher Rijndael

- In recent years it has been shown that the algorithm is not as strong as it was first thought

- If an adversary tries to break the algorithm by searching the key space, then he/she would need to try 2^{100} keys – instead of 2^{128}.

Other Block Ciphers

Algorithm	Key Size	Number of Rounds	Mathematical operations	Applications
IDEA	128 bits	8	XOR, addition, multiplication	PGP
Blowfish	Variable to 448 bits	16	XOR, variable S-boxes, rotation	
RC5	Variable to 2048 bits	Variable to 255	Addition, subtraction, XOR, rotation	
CAST - 128	40 to 128 bits	16	Addition, subtraction, XOR, rotation fixed S-boxes	PGP

Modes of Operation

Modes of Operation

- **A Technique for improving the effect of a cryptographic algorithm, or to make it compatible with various applications.**

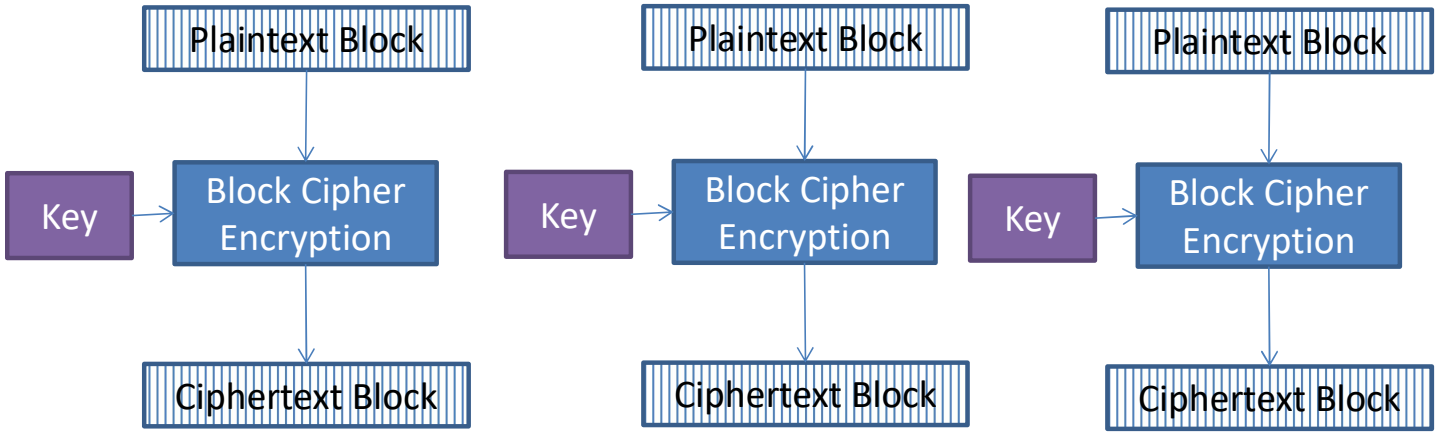
  (Four modes)

- **They enable improving the encryption of block ciphers using the same key.**
  - A block cipher processes one block of data at a time, using the same key.
  - For example, in **DES,** the use of the same key would produce the same ciphertext for similar plain texts. THIS SHOULD BE AVOIDED.
  - This mode of operation is known as “ECB” – See next slide.

Electronic Code Book (ECB)

- A block cipher processes one block of data at a time, using the same key.

- Example:
  - **DES:** This mode of DES algorithm should be avoided as far as possible. Why?



Modes of Operation

- For DEA and TDEA the block length is 64-bits.
- If the same key is used to encrypt the plaintext then there is a unique ciphertext for every 64-bit block of plaintext.
- The codebook is the collection of all the unique plaintext, ciphertext blocks.
- Identical plaintext blocks result in identical ciphertext blocks.
- If the ciphertext is highly structured a cryptanalyst can use these regularities to break the code.

Cipher Block Chaining (CBC)

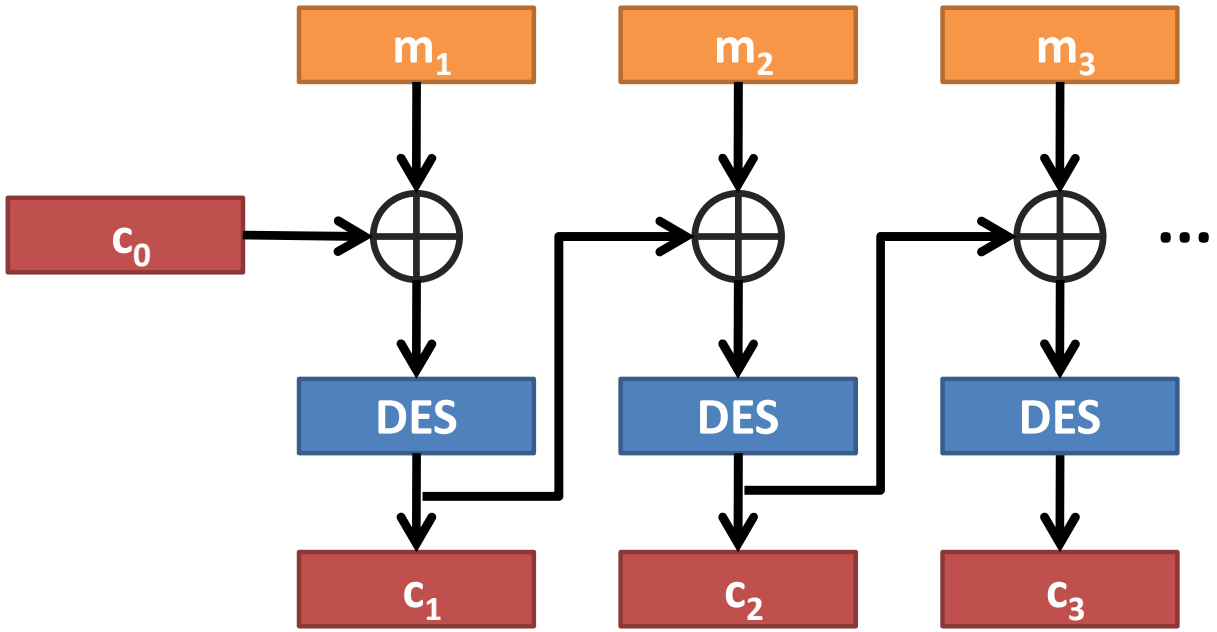
- The encryption depends on the encryption's history. If c_0 is an initialisation block agreed among partners.

- Same as ECB, i.e. One key, but the input is chained to the previous key – making it stronger than before.

Encryption	Decryption
$c_1 = DES(m_1 \oplus c_0)$	$m_1 = DES^{-1}(c_1) \oplus c_0$
$c_2 = DES(m_2 \oplus c_1)$	$m_2 = DES^{-1}(c_2) \oplus c_1$
$c_3 = DES(m_3 \oplus c_2)$	$m_3 = DES^{-1}(c_3) \oplus c_2$
...	...

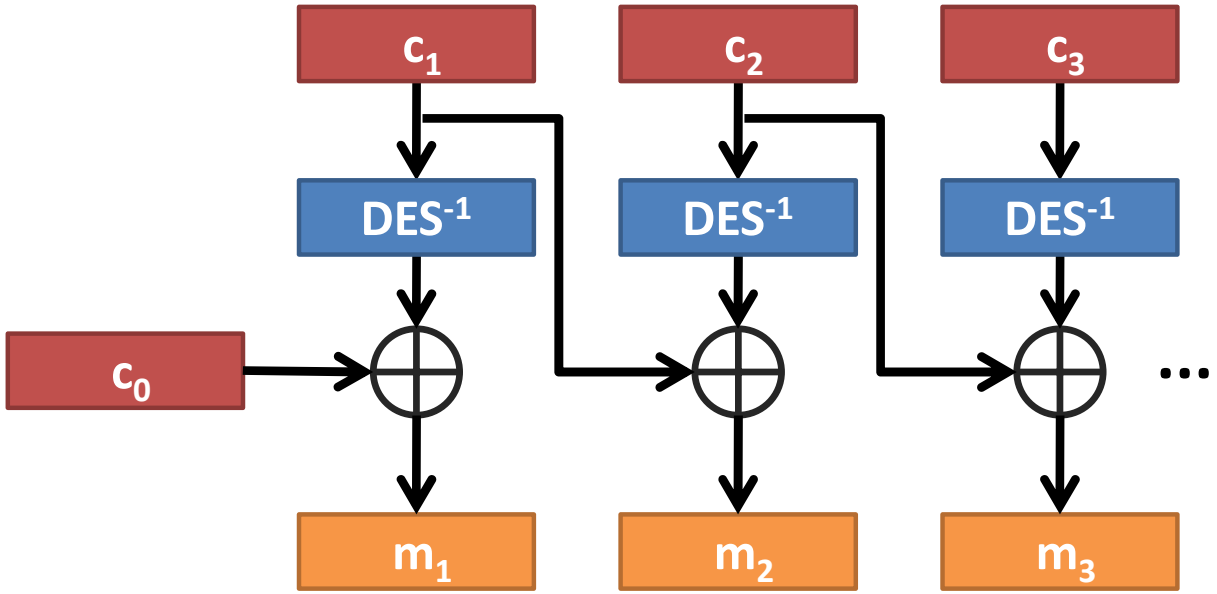
Cipher Block Chaining (CBC)

- **Encryption**



Cipher Block Chaining (CBC)

- **Decryption**



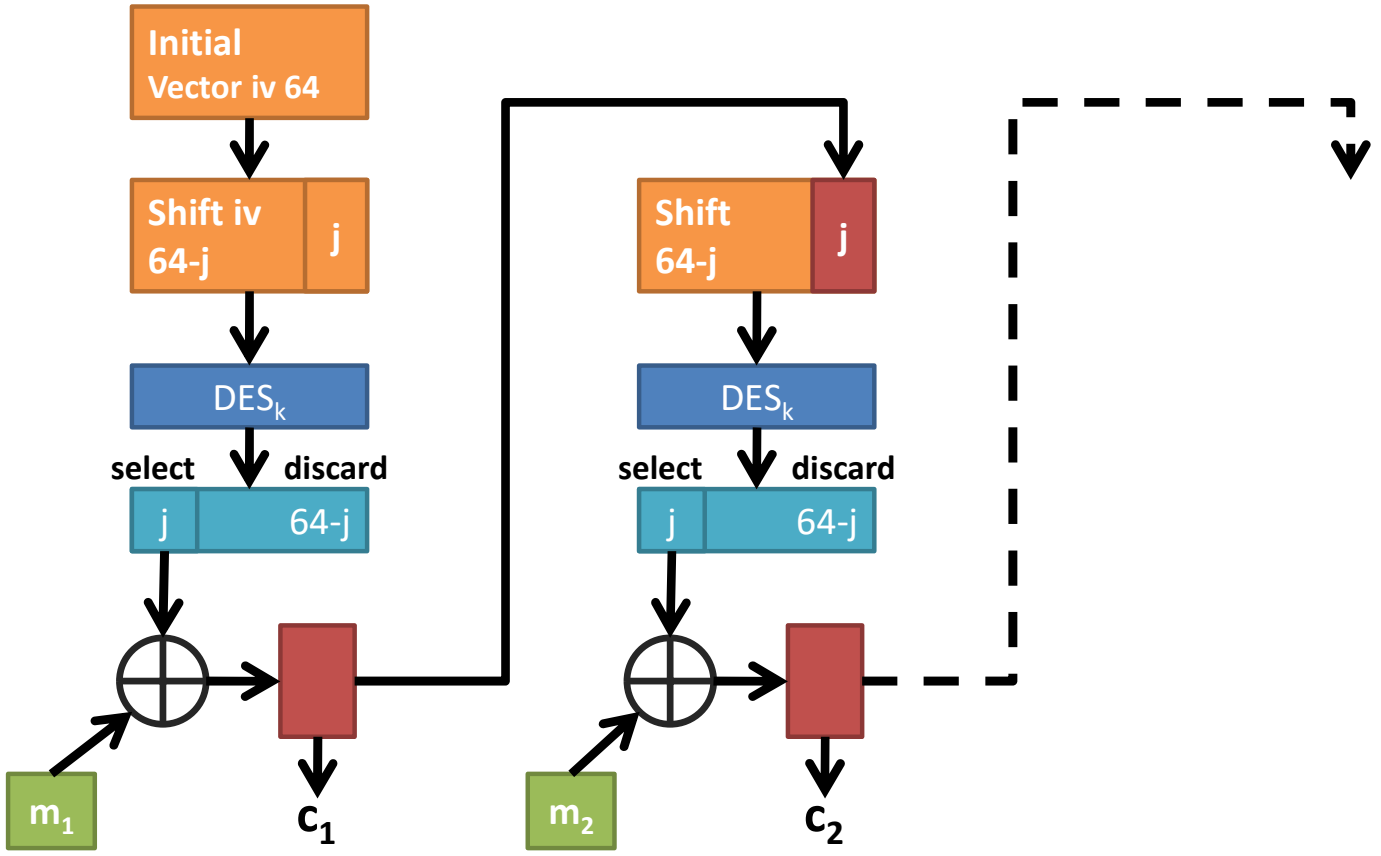
Cipher **f**eed**b**ack (**CFB**) mode

- **CFB converts DES into a stream cipher. If the unit of transmission is j-bits (i.e. j=8 bits)**
  - Start with an initial vector (iv) (given)
  - Shift *j* bits
  - Encrypt using DES
  - Select first *j* bits
  - XOR with the *j* bits of the message
  - Use the encrypted message as new iv

If *j* bit inputs were used, then the output will only need *j* bits ➔ Efficient transmission capacity.

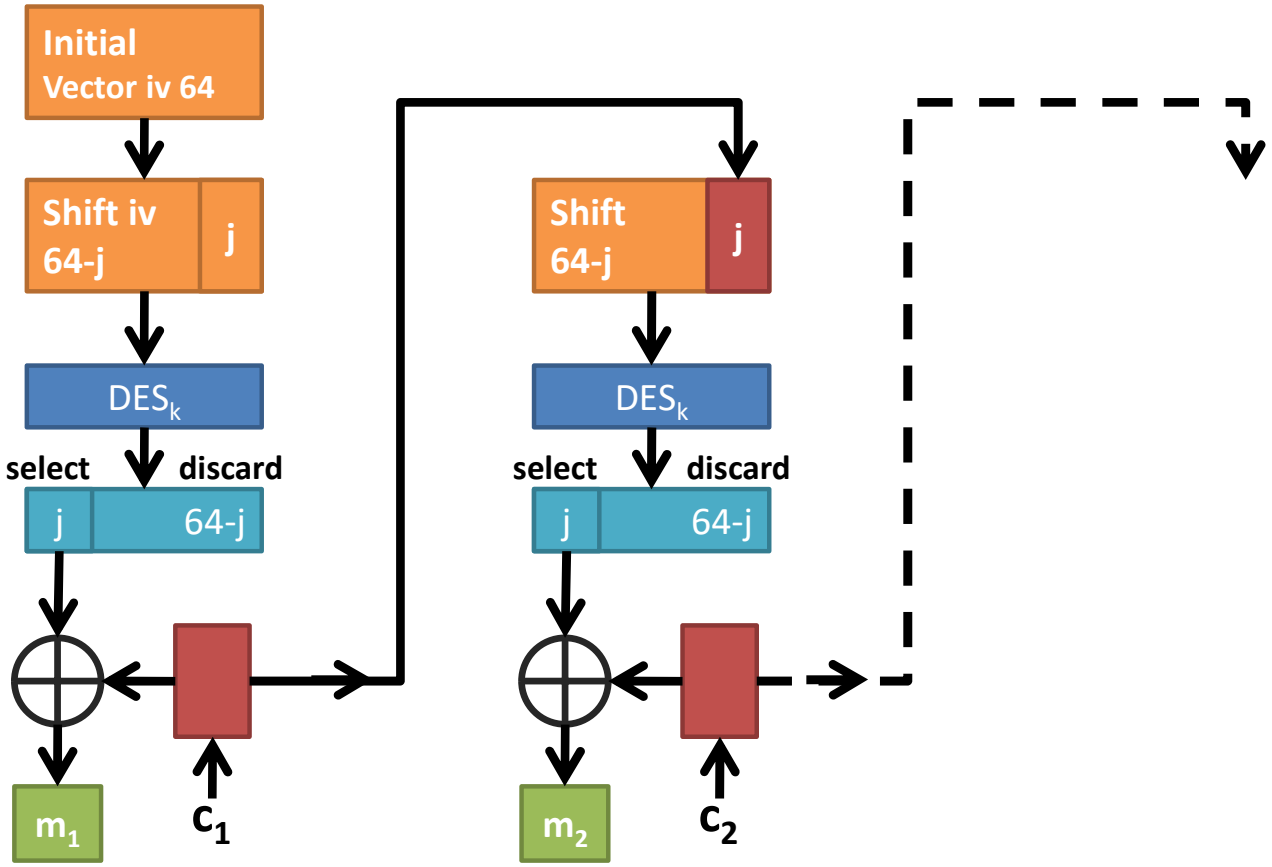
Cipher **f**eed**b**ack (CFB) mode

- **Encryption**



Cipher **f**eed**b**ack (CFB) mode

- **Decryption**



Cipher Feedback mode (CFB)

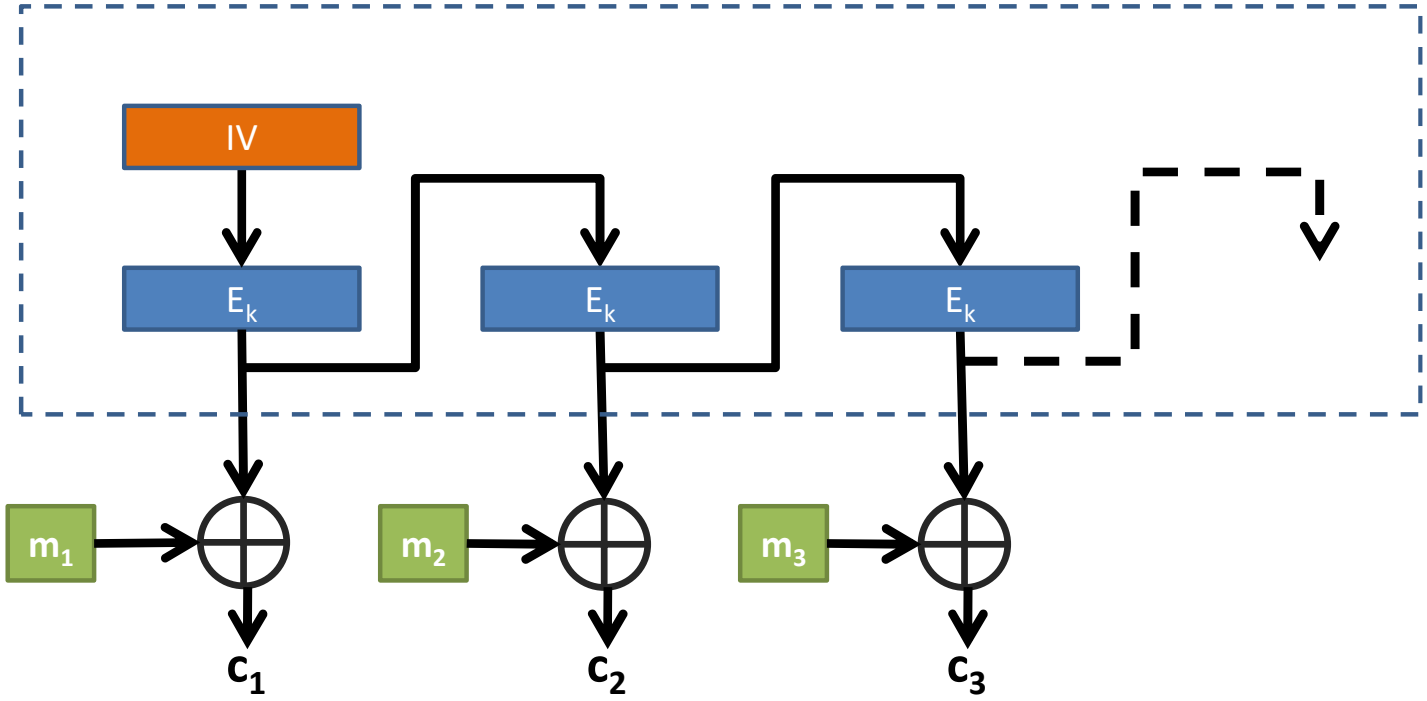
- Error Propagation:
  - For an initial block of 64 bits, if an error bit occurs, it propagates to the next 8 blocks.
    - The reason is that at each step of the CFB shifts the initial value 8 bits, and it would take 8 rounds of CFB to remove the corrupted bits.

Output **f**eed**b**ack mode (**OFB**)

- Unique IV for each use.
- Compared with Cipher Feedback Mode, Output Feedback Mode avoids error propagation.
- Transforms a block cipher into a stream cipher.
- Can be computed in advance.
- Parallel processing is possible:
  - The block cipher operations may be performed in advance, allowing the final step to be performed in parallel once the plaintext or ciphertext is available.

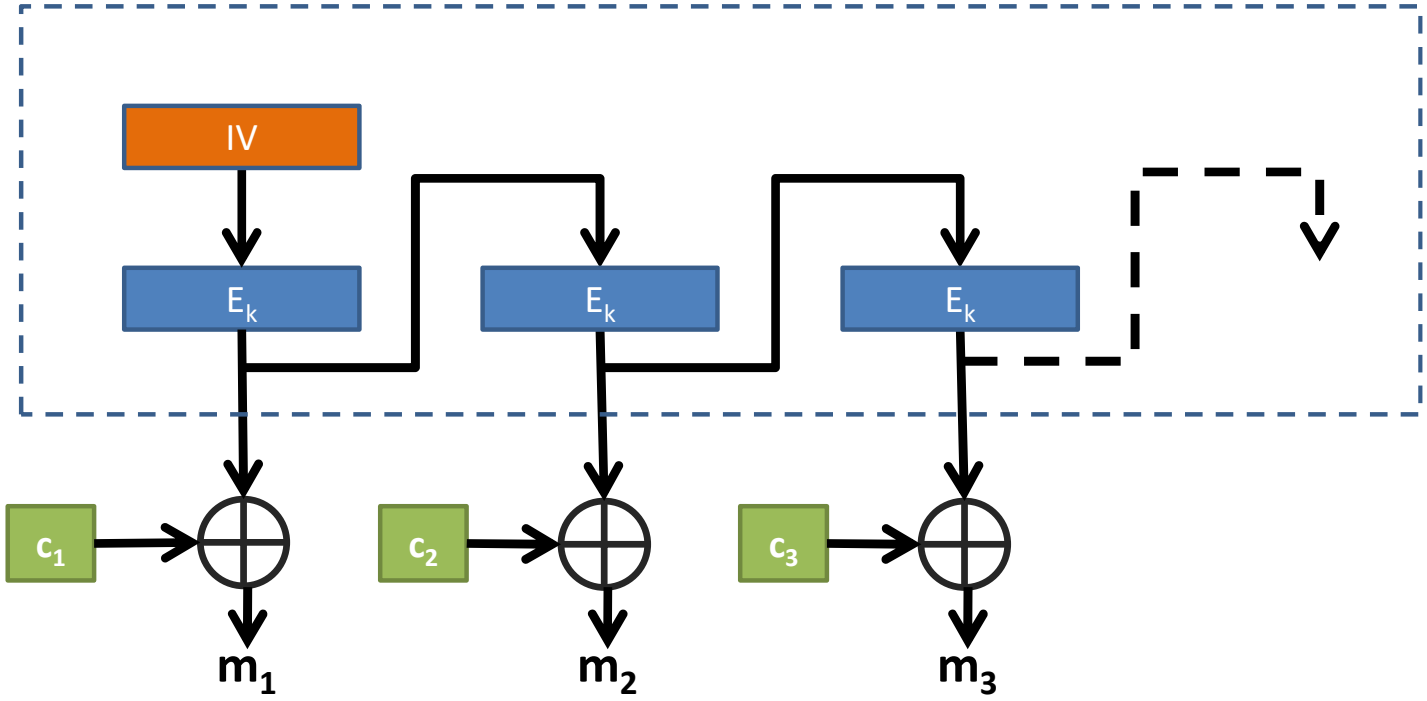
Output **f**eed**b**ack mode (**OFB**)

- Encryption



Output **f**eed**b**ack mode (**OFB**)

- Decryption

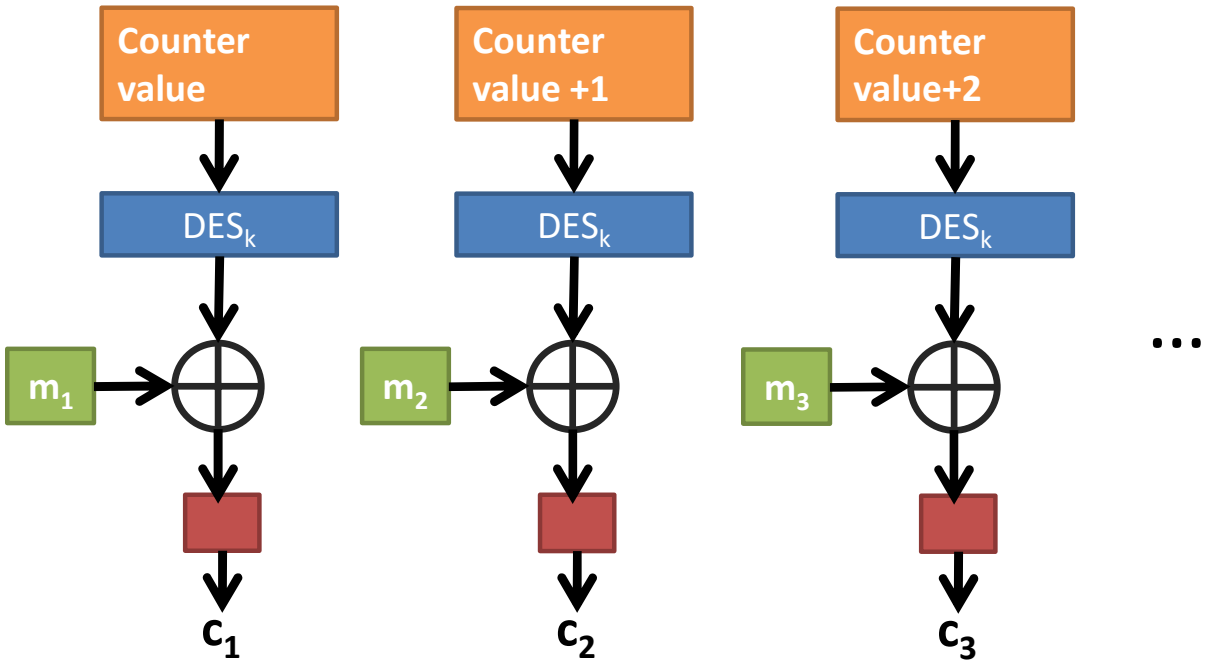


Counter mode (CTR)

- Also known as
  - Integer Counter Mode (ICM)
  - Segmented Integer Counter (SIC) mode
- Like OFB, turns a block cipher into a stream cipher.
- It generates the next keystream block by encrypting successive values of a ‘counter’.
- Increased applications
  - Efficient (HW & SW)
  - Simplicity and Strength

Counter mode (CTR) mode

- **Encryption**



Modes of Operation: summary

Mode	Description	Application
ECB	Block of 64 bits are encoded independently using the same key	- Secure transmission of single values (e.g. Key)
CBC	Input is XOR'ed with the next and previous plaintext ciphertext, respectively.	- Generic block transmission - Authentication
CFB	J bits inputs. Previous ciphertext is used in encryption, then XOR'ed with plaintext.	- General stream transmission - Authentication
OFB	Like CFB, but input is the DES output	- Stream transmission/noisy channel(e.g. Satellite comms.)
CTR	Input XOR'ed with encrypted couter.	- Block transmission - High-speed.

Key Distribution

- The block ciphers security depends on the secrecy of the key.
- The weakest part of all existing crypto systems is the key negotiation. Once the key negotiation is broken the encryption is worthless!

Key Distribution

- **Problems:**
  - There is no message signature. That is a sender cannot prove to his partner that he has sent the message. (e.g. Important problem in E-commerce)
  - The keys have to be negotiated on a channel whose security is higher than the channel used for the normal transmission.
  - The number of keys. For a network with *n* partners that exchange messages with everyone $n(n-1)/2$ keys are needed. (i.e. If $n = 1000$ then number of keys = 999 000)

Key Distribution

- Partner *A* selects the key and physically delivered to partner *B*.
- Third party partner *C* selects the key and physically delivered to *A* and *B*.
- If *A* and *B* have previously and recently used a key, *A* (or *B*) can transmit a new key to the other, encrypted using the old key.
- *A* and *B* have an encrypted connection to *C*, *C* deliver the new key on the encrypted links to *A* and *B*.

Summary

- Conventional Encryption. DES and Rijndael
- Modes of Operation
- Key Distribution

Mode	Description	Application
ECB	Block of 64 bits are encoded independently using the same key	- Secure transmission of single values (e.g. Key)
CBC	Input is XOR'ed with the next and previous plaintext ciphertext, respectively.	- Generic block transmission - Authentication
CFB	J bits inputs. Previous ciphertext is used in encryption, then XOR'ed with plaintext.	- General stream transmission - Authentication
OFB	Like CFB, but input is the DES output	- Stream transmission/noisy channel(e.g. Satellite comms.)
CTR	Input XOR'ed with encrypted couter.	- Block transmission - High-speed.

Additional information

Fields

- Closure of *F* under + and *
  For all *a*,*b belonging to F*, both *a*+*b* and *a*b* belong to *F***
- Both + and * are associative
  For all *a*,*b*,*c* in *F*, *a*+(*b*+*c*) = (*a*+*b*)+*c* and *a(*b***c*) = (*a***b*)**c***
- Both + and * are commutative
  For all *a*,*b* belonging to *F*, *a*+*b* = *b*+*a* and *a*b* = *b***a***
- The operation * is distributive over the **For all**
  For all *a*,*b*,*c* belonging to *F*, *a(*b*+*c*) = (*a***b*)+(*a***c*)**

Fields

- Existence of an additive identity
  There exists an element 0 in *F*, such that for all *a* belonging to *F*, *a*+0 = *a*

- Existence of a multiplicative identity
  There exists an element 1 in *F* different from 0, such that for all *a* belonging to *F*, *a1 = *a***

- Existence of additive inverses
  For every *a* belonging to *F*, there exists an element –*a* such that *a*+(-*a*) = 0

- Existence of multiplicative inverses
  For every $a \neq 0$ to *F*, there exists an element a^{-1} in *F* such that $a*a^{-1} = 1$

$GF(2^8)$

- **Every element of the field is of the form**

$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$

Where b_i is 0 or 1. So $b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$ is the representation of a byte.

- **Example:**

$(x^7+x^6+x^3+x+1) + (x^4+x^3+x^1) = x^7+x^6+x^4+x$

In binary: $11001011 \oplus 00011001 = 11010010$