MIT-WPU

॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

SY BTech Semester-IV (AY 2022-23)

Computer Science and Engineering (Cybersecurity and Forensics)

Disclaimer:

a. Information included in these slides came from multiple sources. We have tried our best to cite the sources. Please refer to the references to learn about the sources, when applicable.

b. The slides should be used only for preparing notes, academic purposes (e.g. in teaching a class), and should not be used for commercial purposes.

Information and Cyber Security (CET3004B)

Examination Scheme: **Credit: 3+1**

Class Continuous Assessment: 30 Marks

Lab Continuous Assessment: 30 Marks

End Semester Examination: 40 Marks

Information and Cyber Security

Course Objectives:

1. Knowledge:

(i) To focus on the models, tools, and techniques for enforcement of security with some emphasis on the use of cryptography. Students will learn security from multiple perspectives

(ii) To educate students on the fundamental principles and techniques of computer and network security

2. Skills:

(i) Acquire background on hash functions, authentication, firewalls, intrusion detection techniques

(ii) Gain hands-on experience with programming and simulation techniques for security protocols

3. Attitude:

(i) Understand the tradeoffs and criteria/concerns for security countermeasure development

(ii) Learn to apply methods for authentication, access control, intrusion detection and prevention

Course Outcomes:

- Analyze and resolve security issues in networks and computer systems to secure an IT infrastructure.
- Apply methods for authentication, access control, intrusion detection and prevention.
- Develop policies and procedures to manage enterprise security risks.
- Evaluate and communicate the human role in security systems with an emphasis on ethics, social engineering vulnerabilities and training.
- Identify software security vulnerabilities, summarize and mitigate security risks associated with integrating systems.

Pre-requisites

- Operating Systems and Computer Networks

Syllabus

Unit: I	**Foundations of Information Security**: Information Security fundamentals, it's need, Confidentiality, Integrity, Availability (CIA triad), Security Policies, Procedures, Guidelines, Standards Administrative Measures and Technical Measures, Attacks, Vulnerability, Security Goals, Security Services and Defense mechanisms **Cryptographic Techniques:** Conventional substitution and transposition ciphers, One-time Pad, Block cipher and Stream Cipher, Cipher modes of operations, Steganography. Symmetric Cryptographic Techniques: DES, AES	**9 Hrs**
Unit: II	**Mathematical Foundations and Public Key Cryptography:** Mathematics for Security: Modular Arithmetic, Euler's theorem, Fermat Theorem, Euclidean Algorithm, Miller-Rabin Algorithm, Primality Test, Chinese Remainder Theorem, Discrete Logarithm, Asymmetric Key Cryptography: RSA algorithms. Hash algorithms: MD5, SHA1	**9 Hrs**

Syllabus (Continue)

Unit: III	**Authentication and Digital Signatures**: Use of Cryptography for authentication, Secure Hash function, Key Management and Distribution: Symmetric Key Distribution, Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys Cryptographic Key Infrastructures, Diffie-Hellman Key Exchange, Digital Certificates x509. Authentication Protocols: Remote, Mutual Authentication, Authentication Methods: Password, Two way methods, Biometric Authentications, Kerberos Security	**9 Hrs**
Unit: IV	**Network and Cyber Security:** Networks Security Fundamentals, Layer-wise Security concerns, Firewalls: Packet filtering, Stateless and Stateful, Intrusion detection systems: host based, network based IDS, Secured Socket Layer Security, IP level IPSEC security, Email Security: PGP, S/MIME. Cyber Security: Definition and origin, Cyber Crime and information security, Types of Cyber Crime, Classification of Cyber Criminals, Tools used in Cyber Crime, Challenges, Strategies, The Legal Perspective-Indian/Global Perspective, Types of Attack, Social Engineering, Cyber stalking, Ransomware.	**9 Hrs**

Syllabus (Continue)

Unit: V	**Cybersecurity Techniques, Tools and Laws:** Introduction, Proxy servers and Anonymizers, Phishing, Password Cracking tools, Key-loggers and Spywares, DoS and DDoS, Viruses, Worms, Trapdoors, Salami attack, Man-in-the- middle attacks, Covert channels, SQL injection, Cyber Security Safeguards- Overview, Access control, Audit, Authentication, Biometrics. Cybercrime and Legal perspectives, Cyber laws Indian context, The Indian IT Act-Challenges, Amendments, Challenges to Indian Law and cybercrime Scenario in India, Indian IT Act and Digital Signatures.	**9 Hrs**

Syllabus (Continue)

Books:- (Reference)	1. Michael E. Whitman and Herbert J. Mattord, “Principles of Information Security”, Cengage Learning; ISBN: 1285448367 2. Christof Paa and Jan Pelzl, “Understanding Cryptography: A Textbook for Students and Practitioners”, Springer; ISBN: 3642041000 3. William Stallings and Lawrie Brown, “Computer Security: Principles and Practice”, Prentice Hall. Swiderski, Frank and Syndex, “Threat Modeling”, Microsoft Press. 4. Ohn W. Rittinghouse, William M. Hancock, “Cyber Security Operations Handbook”, Elsevier Pub. 5. Deborah G Johnson, “Computer Ethics”, 4th Edition, Pearson Education Publication. 6. Earnest A. Kallman, J.P Grillo, “Ethical Decision making and IT: An Introduction with Cases”, McGraw Hill Publication.

Supplementary Reading:

Web Resources: 1. https://www.newhorizons.com/promotions/cybersecurity-ebooks

MOOCs and Weblinks: COURSERA, NPTEL, etc.

- https://nptel.ac.in/courses/106106129
- https://www.udemy.com/course/hands-on-penetration-testing-labs-30/

Assign No.	List of Assignments
1.	Write a program using JAVA or Python or C++ to implement any classical cryptographic technique
2.	Write a program using JAVA or Python or C++ to implement Feistal Cipher structure
3.	Write a program using JAVA or Python or C++ to implement S-AES symmetric key algorithm
4.	Write a program using JAVA or Python or C++ to implement RSA asymmetric key algorithm
5.	Write a program using JAVA or Python or C++ to implement integrity of message using MD5 or SHA
6.	Write a program using JAVA or Python or C++ to implement Diffie Hellman Key Exchange Algorithm
7.	Write a program using JAVA or Python or C++ to implement Digital signature using DSA
8.	Demonstrate Email Security using - PGP or S/MIME for Confidentiality, Authenticity and Integrity
9.	Demonstration of secured web applications system using SSL certificates and its deployment in Apache tomcat server
10.	Configuration and demonstration of Intrusion Detection System using Snort
11.	Configuration and demonstration of NESSUS tool for vulnerability assessment

Guidelines for CCA and LCA

CCA and LCA Marks Distribution		
	Examination Scheme	**Marks**
Class Continuous Assessment (CCA)	Mid-Term Theory Exam	15
	Component 1 (Active Learning)	10
	Component 2 (Theory Assignment)	05
Laboratory Continuous Assessment (LCA)	Practical Performance	10
	Active Learning/Additional implementation/ Mini Project/On paper design	10
	End term practical/Oral Examination	10
End Term	End Term Theory Examination	40
Total		**100**

Unit-I

Foundations of Information Security & Cryptographic Techniques

Foundations of Information Security

- **Cyber security** or **information security** are the techniques of protecting computers, networks, programs and data from unauthorized access or attacks that are aimed for exploitation.

Benjamin Franklin once said

Three people can keep a secret…....

…… if two of them are dead!

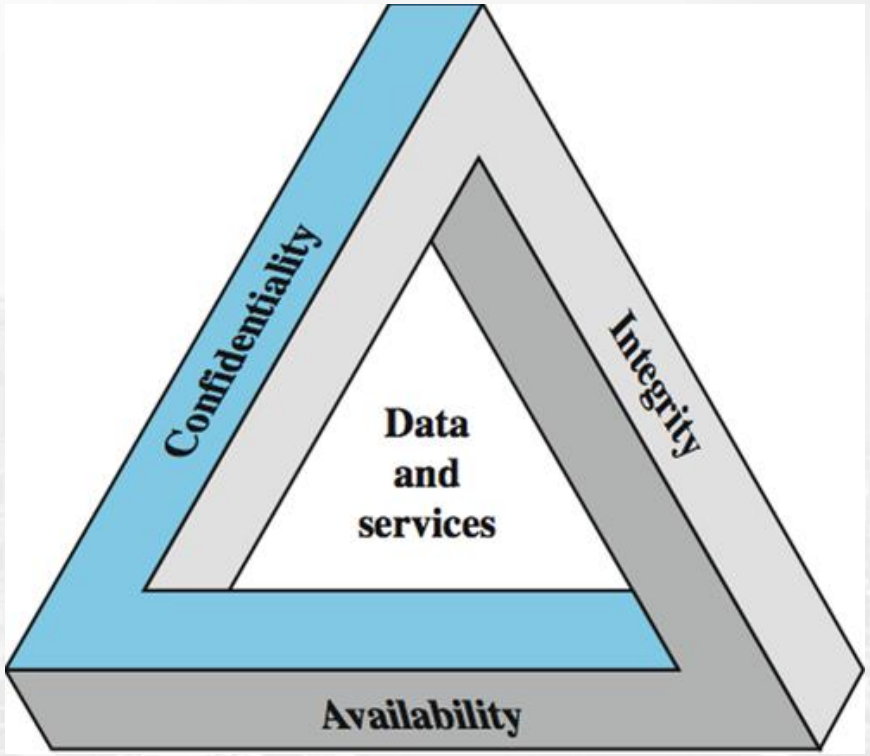
Security is Not Easy to Achieve:

- Human tendency
- Problems of storage and communication
- Trust in all the parties

Key Security Concepts

❑ Elements of Information Security

- ❖ Confidentiality: authorised user can access data
- ❖ Integrity: validity of data
- ❖ Availability



Don’t forget these roots !!

Attacks -- Services -- Defense

?

Security Policies, Procedures, Guidelines…

What is a security policy?

- A security policy is a document that states in writing how a company plans to protect its physical and information technology (IT) assets.
- Security policies are living documents that are continuously updated and changing as technologies, vulnerabilities and security requirements change.
- A company's security policy may include an acceptable use policy. These describe how the company plans to educate its employees about protecting the company's assets. They also include an explanation of how security measurements will be carried out and enforced, and a procedure for evaluating the effectiveness of the policy to ensure that necessary corrections are made.

Security Policies, Procedures, Guidelines…

Why are security policies important?

- Security policies are important because they protect an organizations' assets, both physical and digital. They identify all company assets and all threats to those assets.
- Physical security policies are aimed at protecting a company's physical assets, such as buildings and equipment, including computers and other IT equipment.
- Data security policies protect intellectual property from costly events, like data breaches and data leaks.

Policy:

Set of detailed rules as to what is allowed on the system and what is not allowed.

- User Policy
- System Policy
- Network Policy
- US Law
- Trust

User-level Policy

- Authentication: Method, Protection, Disclosure
- Importing software: Process, Safeguards, Location
- File protection: Default, Variations
- Equipment management: Process, Physical Security
- Backups: How, When
- Problem reporting: Who, How, Emergencies

System-level Policy

- Default configuration
- Installed Software
- Backups
- Logging
- Auditing
- Updates
- Principle servers or clients

Network-level Policy

- Supported services
- Exported services: Authentication, Protection, Restriction
- Imported services: Authentication, Protection, Privacy
- Network security mechanisms

US Law

- General advice - not legal counsel
- Before performing legal actions -- consult a lawyer!
- Legal Options
- Legal Hazards
- Being the target of an investigation
- General Tips
- Civil Actions
- Intellectual Property
- Liability

Trust

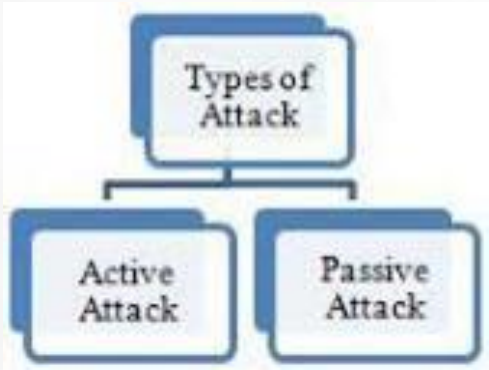
- Tools of computer security are resident on computers
- Just as mutable as any other information on computers
- Can we trust our computer?
- Can we trust our software?
- Can we trust our suppliers?
- Can we trust our people?
- Trust, but verify

Aspects of Security

- consider 3 aspects of information security:
  - **security attack:** Any action that compromises the security of information owned by an organization.
  - **security mechanism:** A process that is designed to detect, prevent, or recover from a security attack.
  - **security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization.

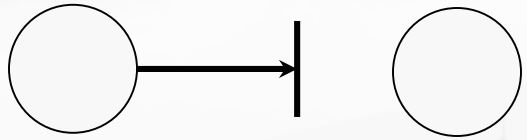
- note terms
  - ***threat:*** a potential for violation of security
  - ***attack:*** an assault on system security, a deliberate attempt to evade security services

Security Attacks - Security threats

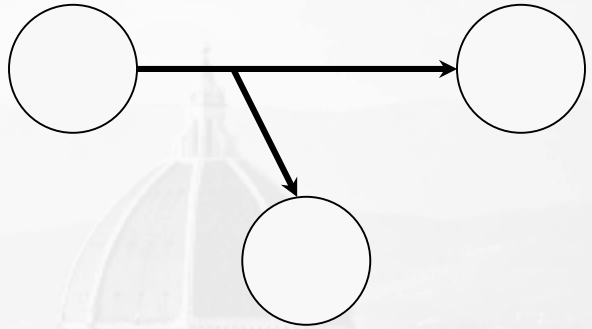


- Interruption – attack on availability
- Interception – attack on confidentiality
- Modification – attack on integrity
- Fabrication – attack on authenticity
  e.g. Email Spoofing, SQL Injection

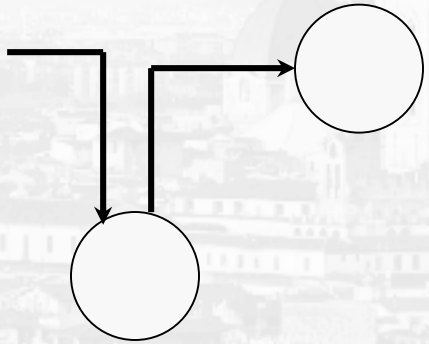
a) Normal flow



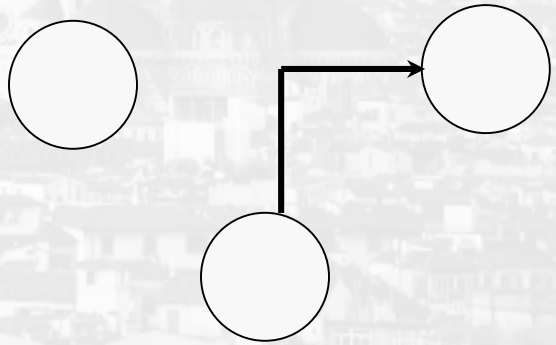
b) Interruption



c) Interception

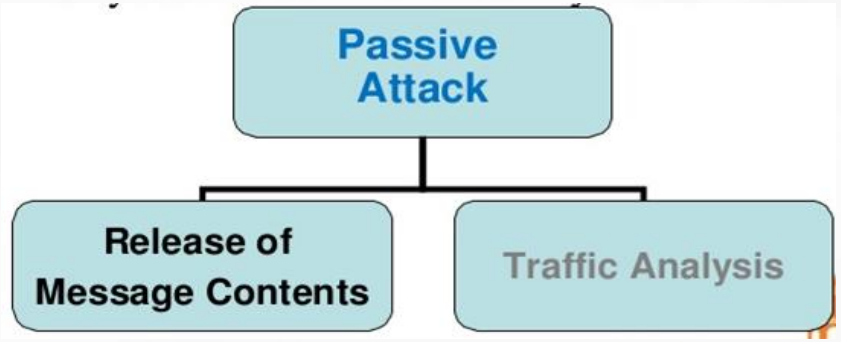


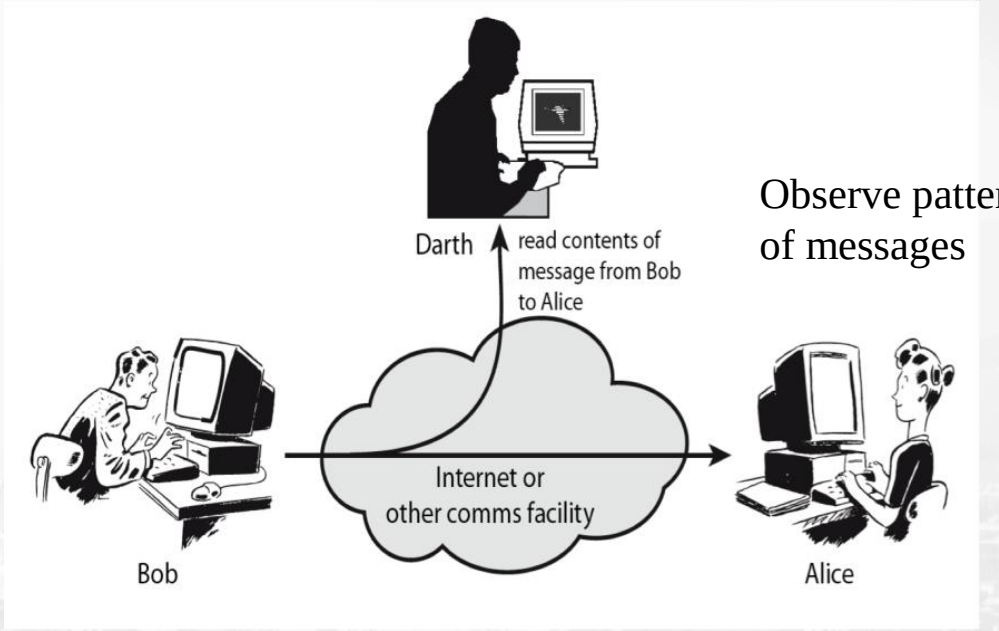
d) Modification



e) Fabrication

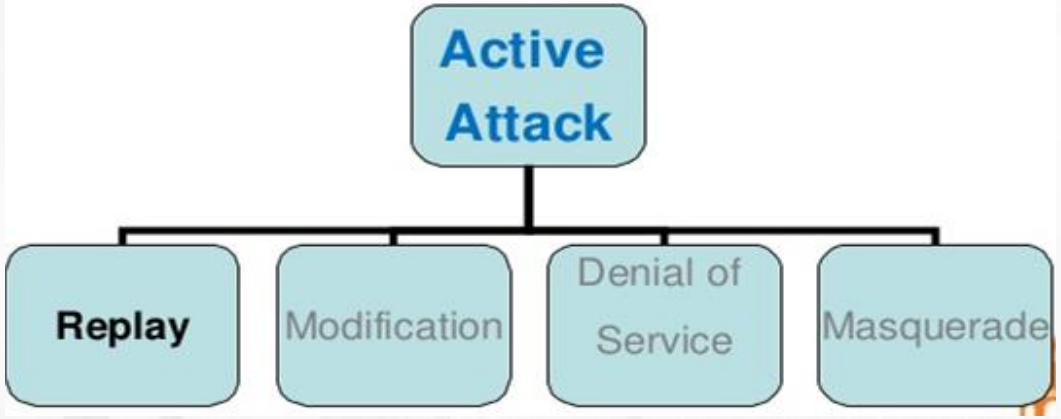
- Passive Attack: make use of information from the system but does not affect system resource

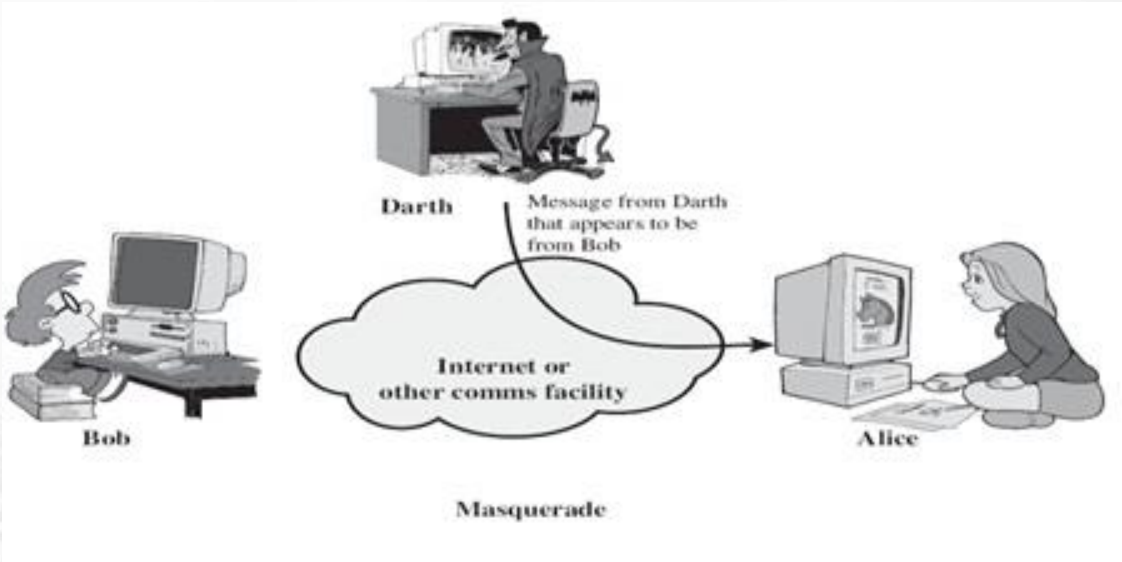




Note: in dealing with passive attacks is on prevention rather than detection. i.e. encryption

- Active Attack: modification of the data stream or the creation of a false stream





Attackers

Adversary	Goal
Student	To have fun snooping on people's e-mail
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by e-mail
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military or industrial secrets
Terrorist	To steal germ warfare secrets

Security Services/Goals

- Confidentiality (privacy)
- Authentication (who created or sent the data)
- Integrity (has not been altered)
- Non-repudiation (prevents either sender or receiver from denying a transmitted message)
- Access control (prevent misuse of resources)
- Availability (permanence, non-erasure)

Security Mechanism

- feature designed to detect, prevent, or recover from a security attack
- no single mechanism that will support all services required
- however one particular element underlies many of the security mechanisms in use:
  - **cryptographic techniques**

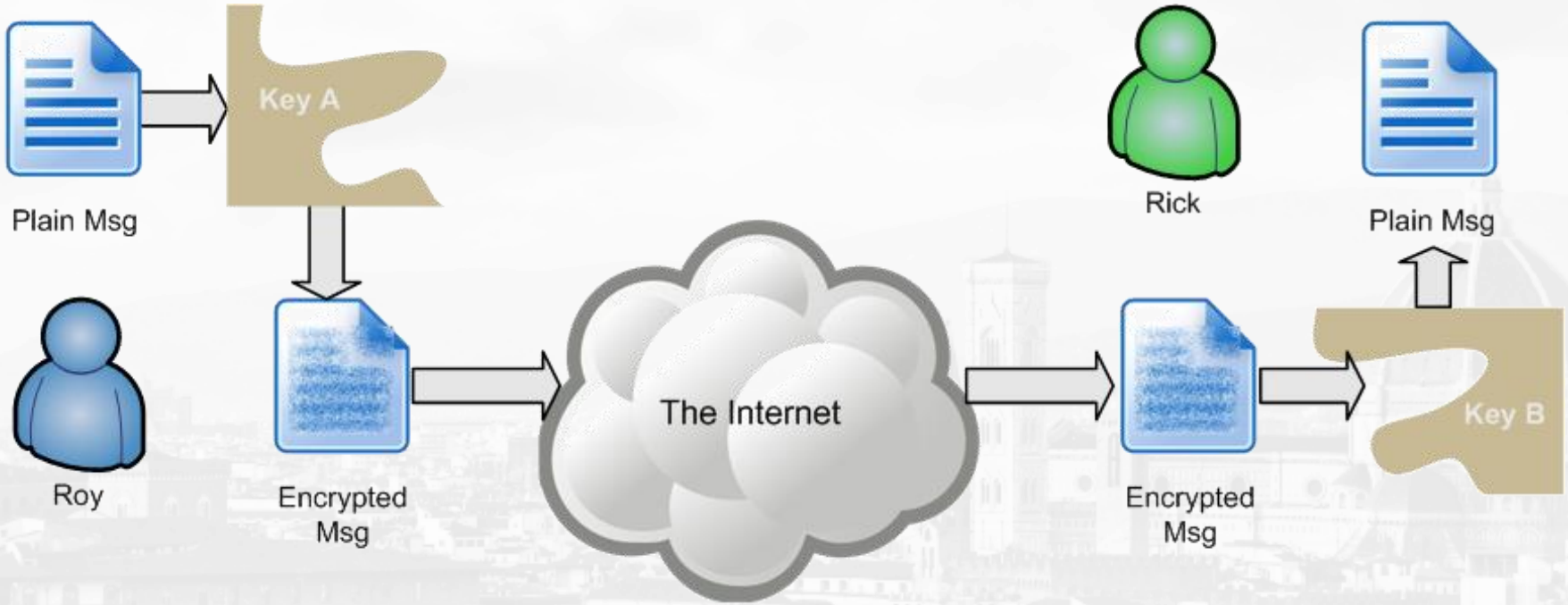
- **specific security mechanisms:**
  - encipherment, digital signatures, access controls, data integrity, authentication exchange, traffic padding, routing control, notarization
- **pervasive security mechanisms:**
  - trusted functionality, security labels, event detection, security audit trails, security recovery

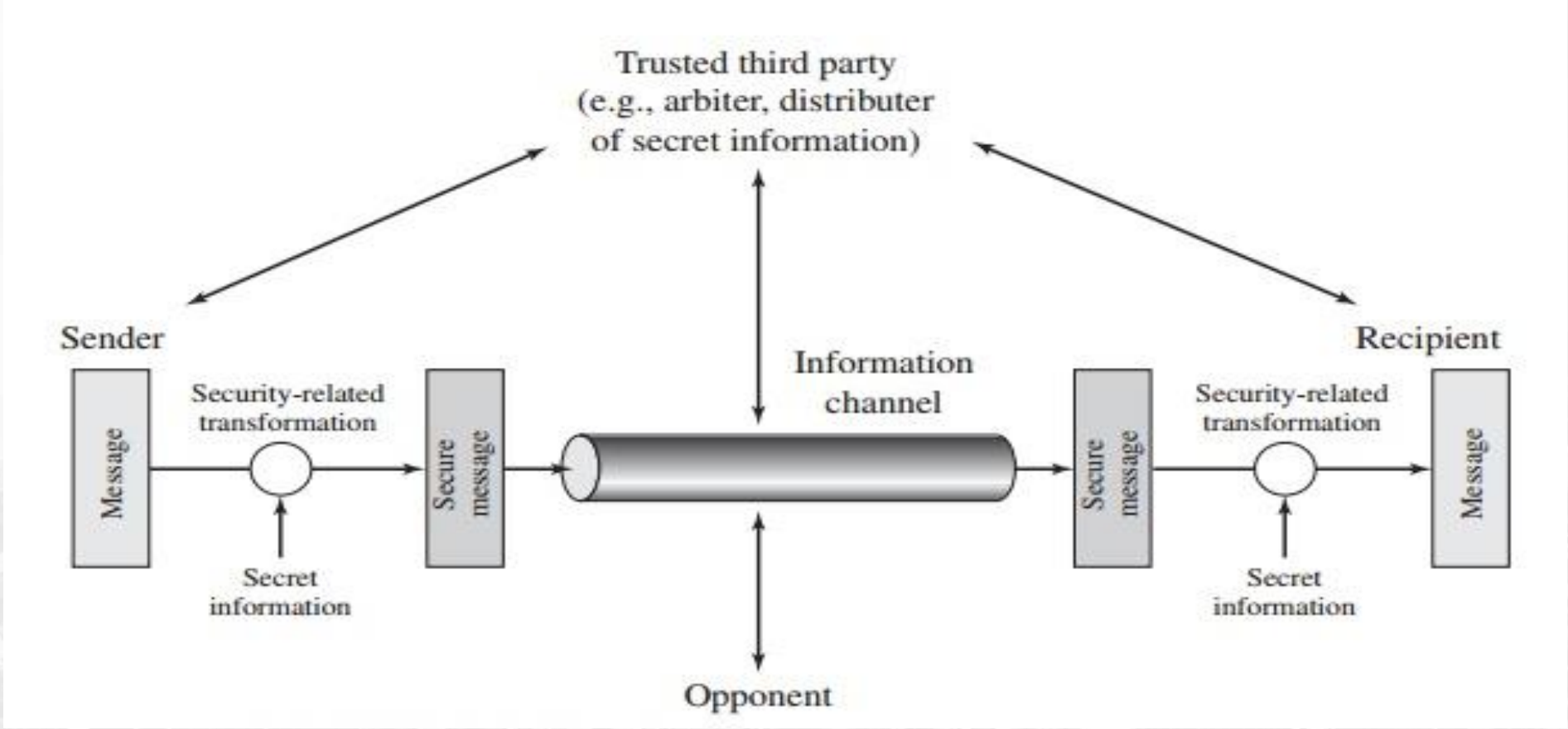
Quiz : Match the Following?

a)Interruption	1)integrity
b)Interception	2)availability
c)Modification	3)authentication
d)Fabrication	4)confidentiality

The Operational Model of Network Security

- Prevention is better than cure




Trusted third party
(e.g., arbiter, distributer
of secret information)
Sender
Recipient
Information
channel
Message
Security-related
transformation
Secure
message
Secure
message
Security-related
transformation
Message
Secret
information
Secret
information
Opponent

Problem

Problem 1: Consider an automated teller machine (ATM) in which users provide a personal identification number (PIN) and a card for account access. Give examples of confidentiality, integrity, and availability requirements associated with the system and, in each case, indicate the degree of importance of the requirement.

Solution: The system must keep personal identification number (PIN) confidential, both in the host system and during transmission for a transaction. In addition, for security the personal identification number must encrypted.

It must protect the integrity of account records and of individual transactions.

Availability of the host system is important to the economic well being of the bank, but not to its fiduciary responsibility. The availability of individual teller machines is of less concern.

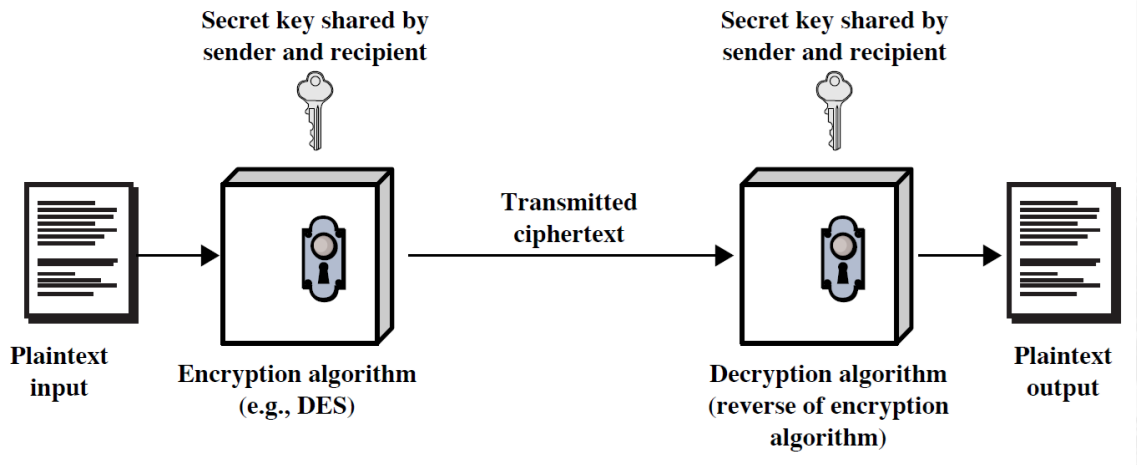
Classical Cryptography

Basic Terminology

- Plaintext- the original message
- Ciphertext - the coded message
- Key - info used in cipher known only to sender/receiver
- Encipher (encrypt) - converting plaintext to ciphertext
- Decipher (decrypt) - recovering ciphertext from plaintext
- Cryptography - study of encryption principles/methods
- Cryptanalysis (codebreaking) - the study of principles/ methods of deciphering ciphertext without knowing key
- cryptology - the field of both cryptography and cryptanalysis

Encryption Methods

- **Symmetric encryption-** DES, Triple DES, AES
- **Asymmetric encryption-** RSA, ECC
  - The security of encryption algorithm depends upon the key



- Symmetric encryption or conventional / private-key/ single-key
- sender and recipient share a common key
- all classical encryption algorithms are private-key
- was only type prior to invention of public-key in 1970's

Cryptography

-- Parameters used by cryptographic systems are:

- The type of **operations** used for transforming plaintext to ciphertext
  - e.g. substitution and transposition
- The number of **keys** used e.g. symmetric, asymmetric
- The way in which the plaintext is **processed** e.g. block cipher, stream cipher

❑ Substitution Ciphers:

Classical Ciphers:

- Plaintext is viewed as a sequence of elements (e.g., bits or characters)
- Substitution cipher : replacing each element of the plaintext with another element.
- Transposition (or permutation) cipher : rearranging the order of the elements of the plaintext.
- Product cipher : using multiple stages of substitutions and transpositions

Caesar Cipher

- Earliest known substitution cipher. Invented by Julius Caesar
- Each letter is replaced by the letter **three** positions further down the alphabet.

Plain: 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: 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: **mit pune** ⟶ **PLW SXQH**
  - Mathematically, map letters to numbers:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

E.g. break ciphertext using shift 2 "**GCUA VQ DTGCM**"
Answer: easy to break

- Then the general Caesar cipher is:

$$c = E_K(p) = (p + k) \bmod 26$$

$$p = D_K(c) = (c - k) \bmod 26$$

❖ Brute-force attack: tries every possible key

Monoalphabetic Cipher

- Shuffle the letters and map each plaintext letter to a different random ciphertext letter:

  Plain letters: abcdefghijklmnopqrstuvwxyz

  Cipher letters: DKVQFIBJWPESCXHTMYAUOLRGZN

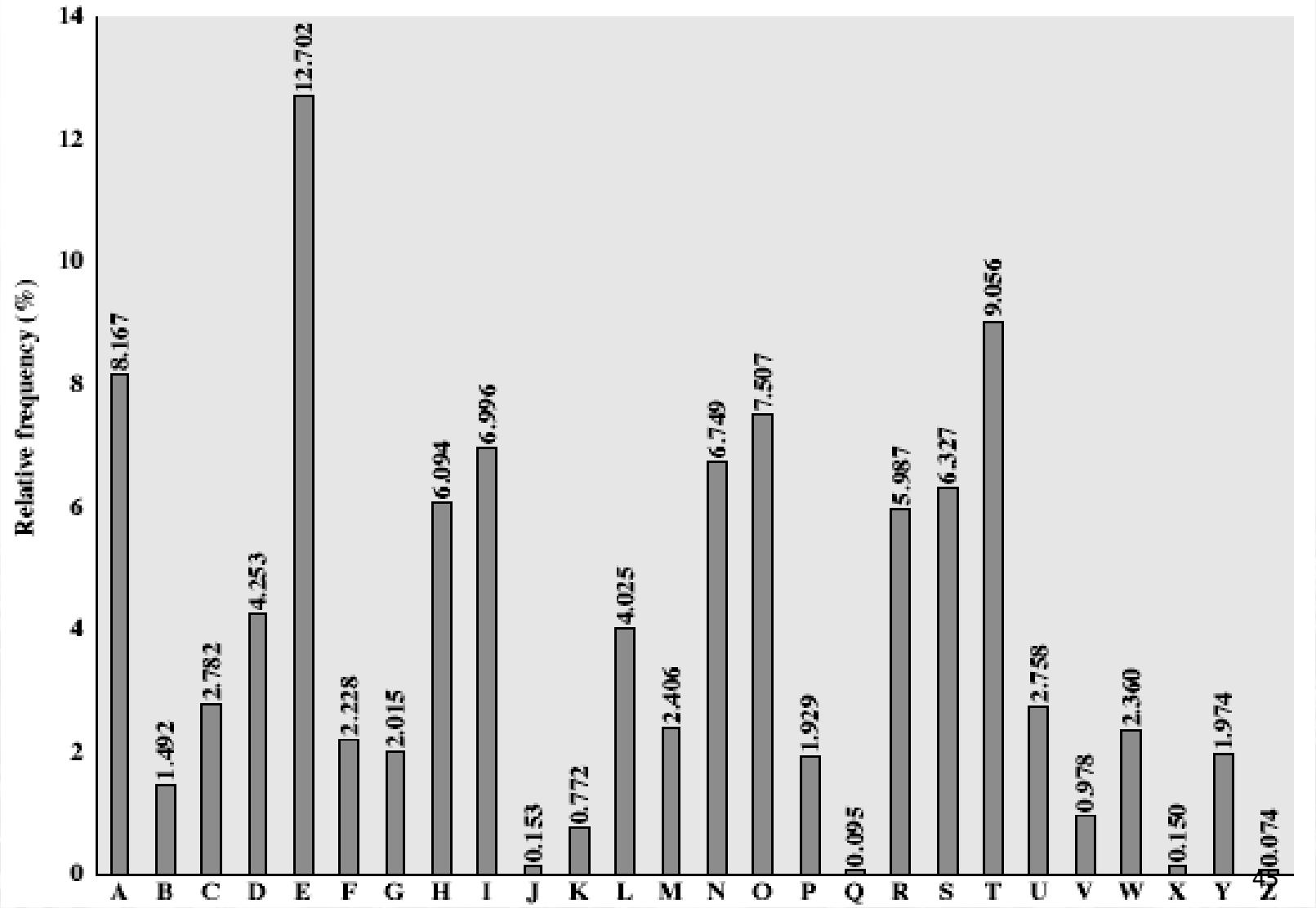
  Plaintext: ifwewishtoreplaceletters

  Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA

- Now we have a total of **$26! = 4 \times 10^{26}$ keys**.
- With so many keys, it is secure against brute-force attacks.
- But not secure against some cryptanalytic attacks.
- Problem is language characteristics.

English Letter Frequencies

- Relative frequency of letters can be determined
- frequency of two-letter combination



- given ciphertext:
  UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
  VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
  EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

- count relative letter frequencies (see text)
- guess **P & Z** are **e and t**
- guess **ZW** is **th** and hence **ZWP** is **the**

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
 t a         e  e te  a that e e a        a
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
   e t   ta t ha e ee  a e  th    t  a
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
 e  e e tat  e   the   t
```

- proceeding with trial and error finally get:
  it was disclosed yesterday that several informal but
  direct contacts have been made with political
  representatives of the Viet cong in Moscow

One-Time Pad

- The number of possible keys is equal to the number of possible plaintexts
- The key is selected at random from the choice of all possible keys
- Any key should only be used once
- It is unbreakable since ciphertext bears no statistical relationship to the plaintext

Plaintext:	M	E	E	T	M	E	O	U	T	S	I	D	E
Numerical Plaintext:	12	4	4	19	12	4	14	20	19	18	8	3	4
OTP:	B	D	U	F	G	H	W	E	I	U	F	G	W
Numerical OTP:	1	3	20	5	6	7	22	4	8	20	5	6	22
Numerical Ciphertext:	13	7	24	24	18	11	10	24	1	12	13	9	
Ciphertext:	N	H	Y	Y	S	L	K	Y	B	M	N	J	

4 + 22 = 0 modular 26

$c_i = p_i \text{ XOR } k_i$

Two fundamental difficulties:

- Problem of making large quantities of random keys
- Problem of key distribution and protection

❖ Because of these difficulties, the one-time pad is of limited utility and is useful primarily for low-bandwidth channels requiring very high security.

Transposition Cipher

- The order of alphabets in the plaintext is rearranged to form a cipher text.
- The **Rail Fence cipher** is a form of transposition cipher
- Plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.
- The example message is: "meet me after the toga party" with a rail fence of depth 2.

```
m e m a t r h t g p r y
  e t e f e t e o a a t
```

Ciphertext:
MEMATRHTGPRYETEFETEOAAT

Row Transposition Cipher

- Write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns
- The order of the columns then becomes the key to the algorithm
- Plaintext: attack postponed until two am
- Key: 4 3 1 2 5 6 7

Key: **4 3 1 2 5 6 7**
Plaintext: **a t t a c k p**
o s t p o n e
d u n t i l t
w o a m x y z

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

Transposition Cipher (2)

- Plaintext written in a row under the key and then arrange the column as per alphabetical order.

❖ Single Columnar Transposition

Preparing the Key:

- Numbered each letter of the key as per their appearance in the alphabet

h	e	a	v	e	n
4	2	1	6	3	5

Preparing the Plaintext: we are the best

h	e	a	v	e	n
4	2	1	6	3	5
W	E	A	R	E	T
H	E	B	E	S	T

Transposition Cipher(3)

Encryption:

a	e	e	h	n	v
1	2	3	4	5	6
A	E	E	W	T	R
B	E	S	H	T	E

Decryption: ABEEESWHTTRE

h	e	a	v	e	n
4	2	1	6	3	5
W	**E**	**A**	R	**E**	T
H	**E**	**B**	E	**S**	T

Problem: Using Transposition cipher encrypt message "WE ARE THE BEST" use key 'HEAVEN'

Transposition Ciphers(4)

❖ Double Columnar Transposition

h	e	a	v	e	n
4	2	1	6	3	5
W	E	A	R	E	T
H	E	B	E	S	T

ABEEESWHTTRE

a	n	o	t	h	e	r
1	4	5	7	3	2	6
A	B	E	E	E	S	W
H	T	T	R	E		

Modes of Operation

- A block cipher processes the data blocks of fixed size.
- Usually, the size of a message is larger than the block size.
- Hence, the long message is divided into a series of sequential message blocks, and the cipher operates on these blocks one at a time.
- The size of plaintext block and ciphertext block is same.
- When no. of bits in the plaintext/message are not multiple of the block size ---- Modes of operation

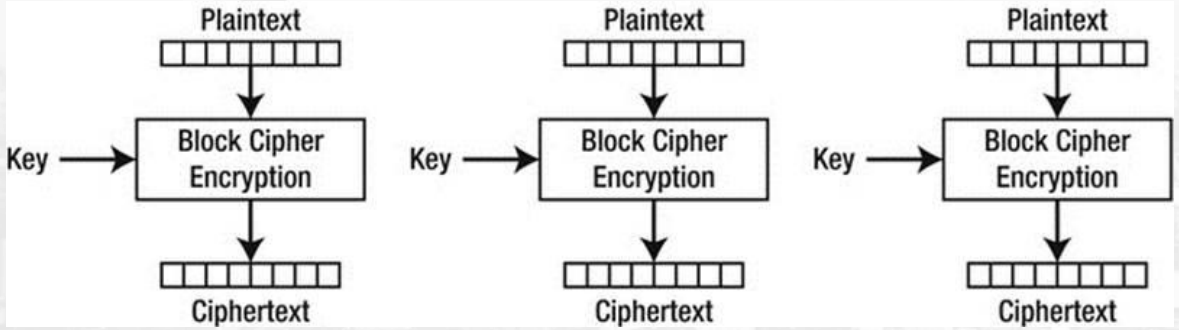
Block Cipher Modes of Operation

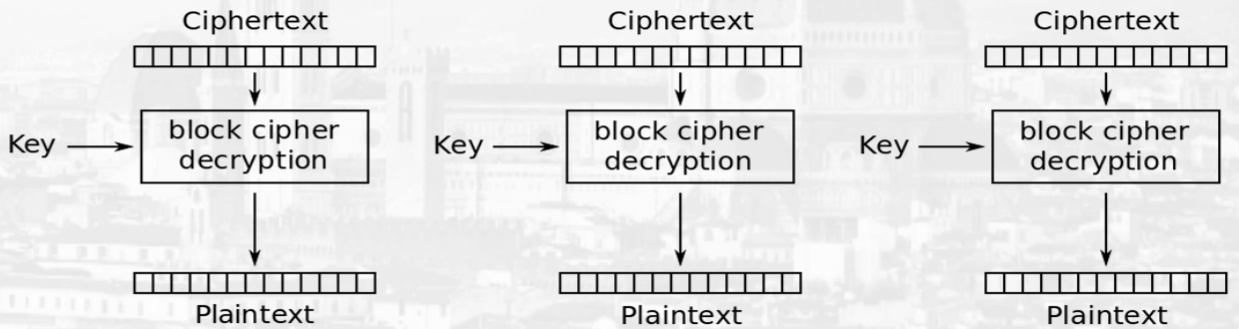
- **Electronic Code book (ECB) mode**
- **Cipher Block Chaining (CBC) mode**
- **Feedback (CFB) modes**
- **Counter (CTR) mode**

Electronic Code book (ECB) mode

- The message is divided into blocks, and each block is encrypted separately.
- If two plaintext blocks are identical then the ciphertext block are also same . Therefore, **a known plaintext attack is possible**.
- **uses**: secure transmission of single values

$$C_i = E_K(P_i)$$

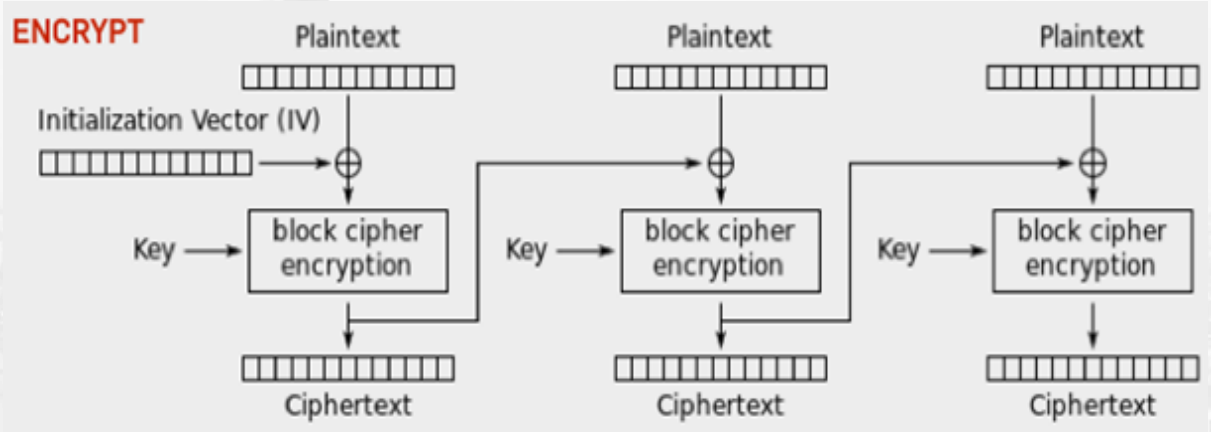


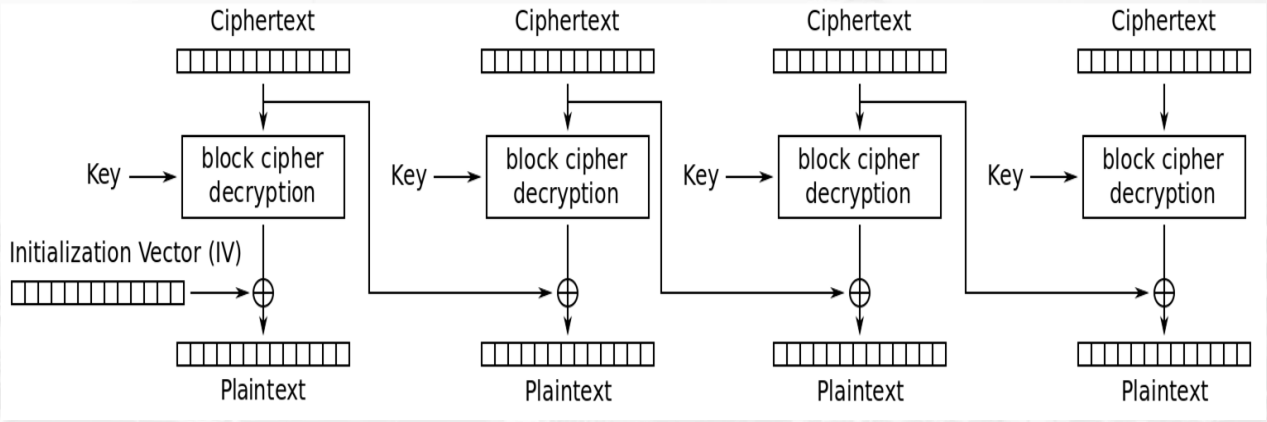


Cipher Block Chaining (CBC) mode

- An initialisation is random number is used to increase security.
- It can be used to generate the hash value.
- **uses:** bulk data encryption, authentication

$$C_i = E_K(P_i \text{ XOR } C_{i-1})$$
$$C_{-1} = IV$$





Advantages:

- For identical block of plaintext, different ciphertext blocks are generated. So secure than ECB mode.
- Hash value, i.e. last ciphertext block, helps to identify if the message is original or modified.

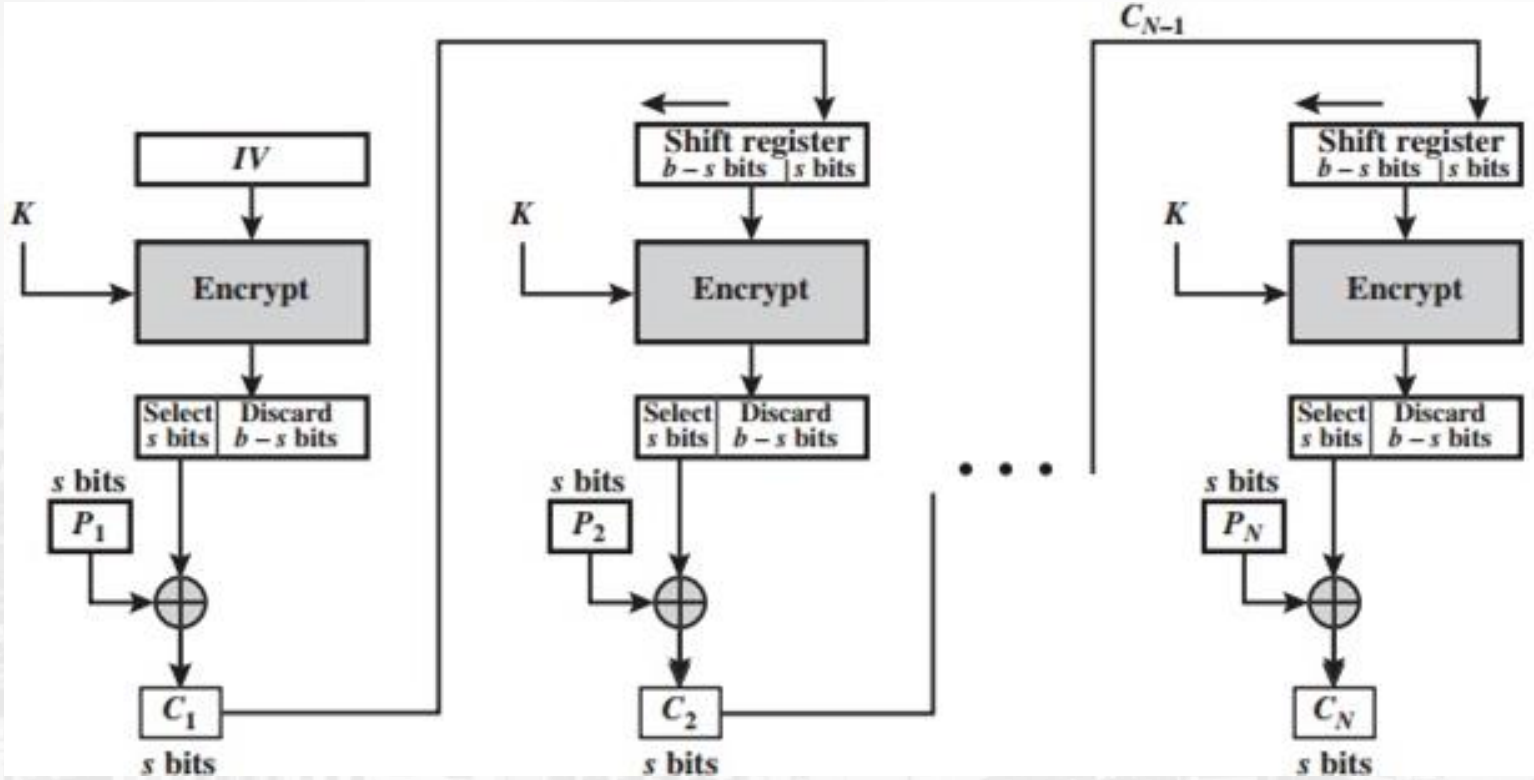
Disadvantages:

- Parallel operation cannot be performed.
- Lost/missing of any block of ciphertext stops the decryption process of the remaining blocks.

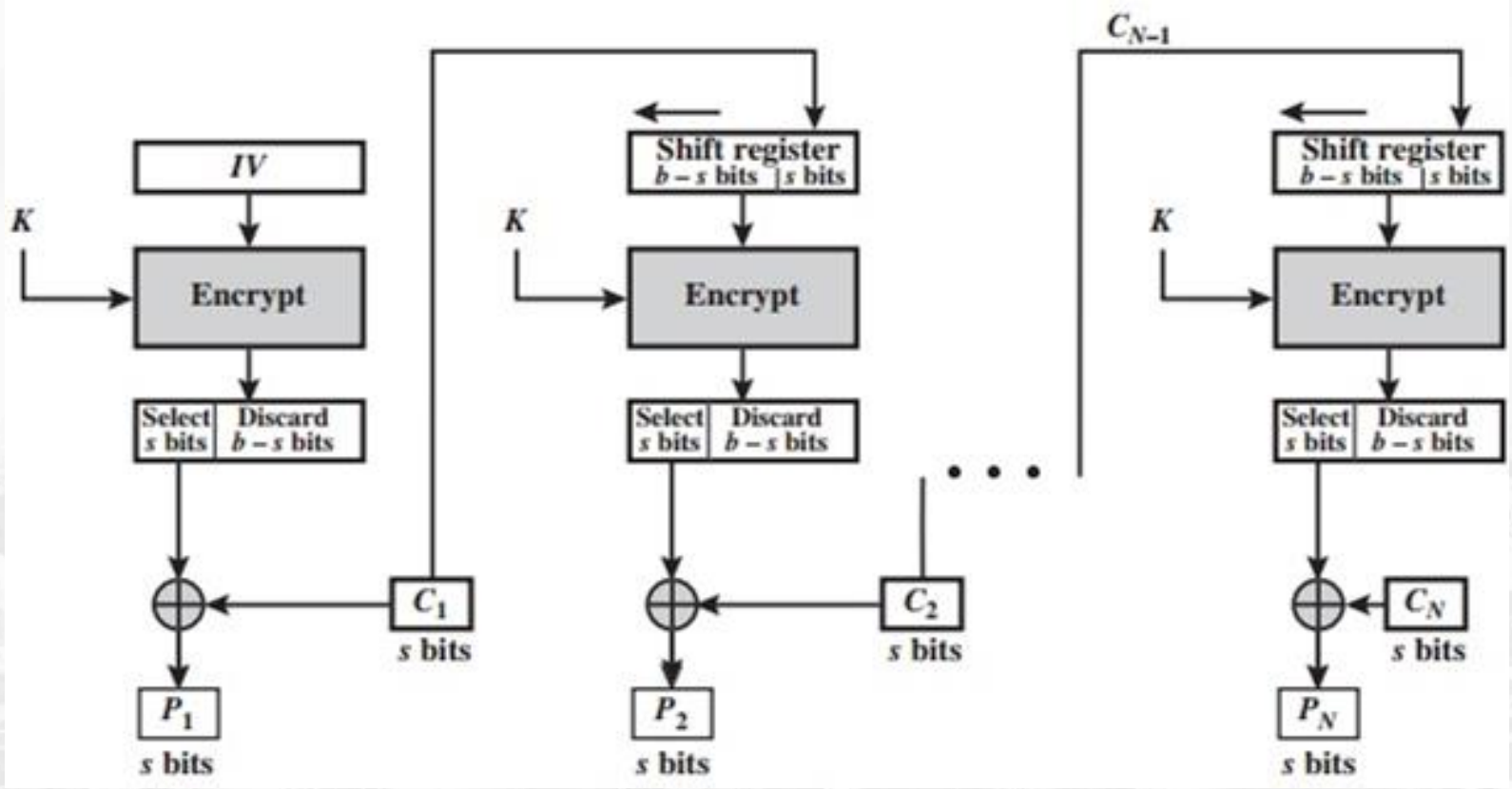
Feedback Mode: Cipher Feedback (CFB) Mode

- Can be used when the block size is smaller than the required block size.
- The block size may be a bit or bytes, so there is no need of padding.
- **uses:** stream data encryption, authentication

$$C_i = P_i \text{ XOR } E_K(C_{i-1})$$
$$C_{-1} = IV$$

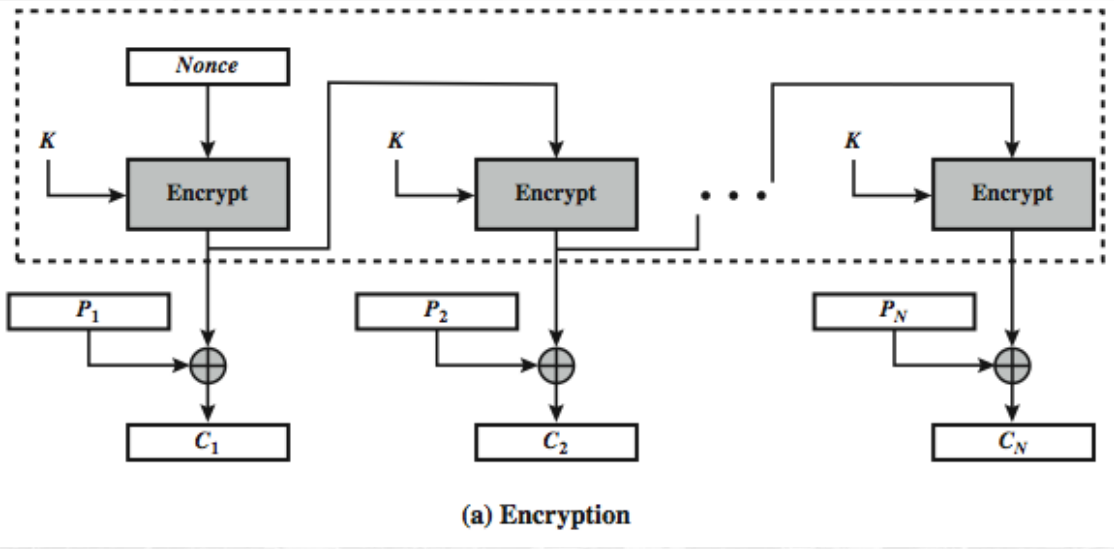


❖ CFB is suffered from bit errors. If in the incoming cipher block, any one bit error is there, then it causes the bit error at the same bit position in the plaintext block.

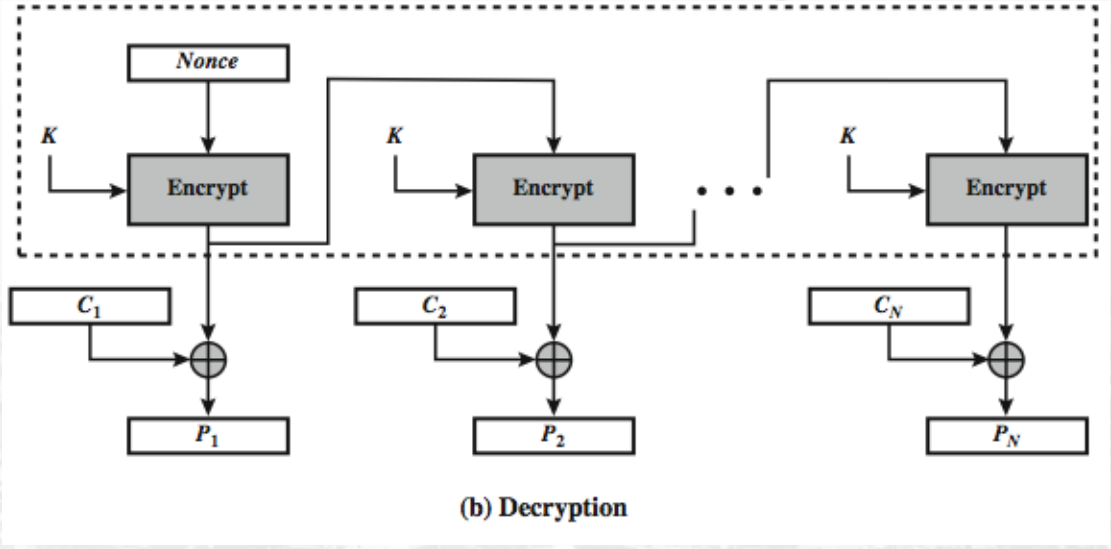


Feedback Mode: Output Feedback (OFB) Mode

- message is treated as a stream of bits
- Free from bit error rate.
- Information about the key is not required, which help the cryptanalyst to break the cipher easily.



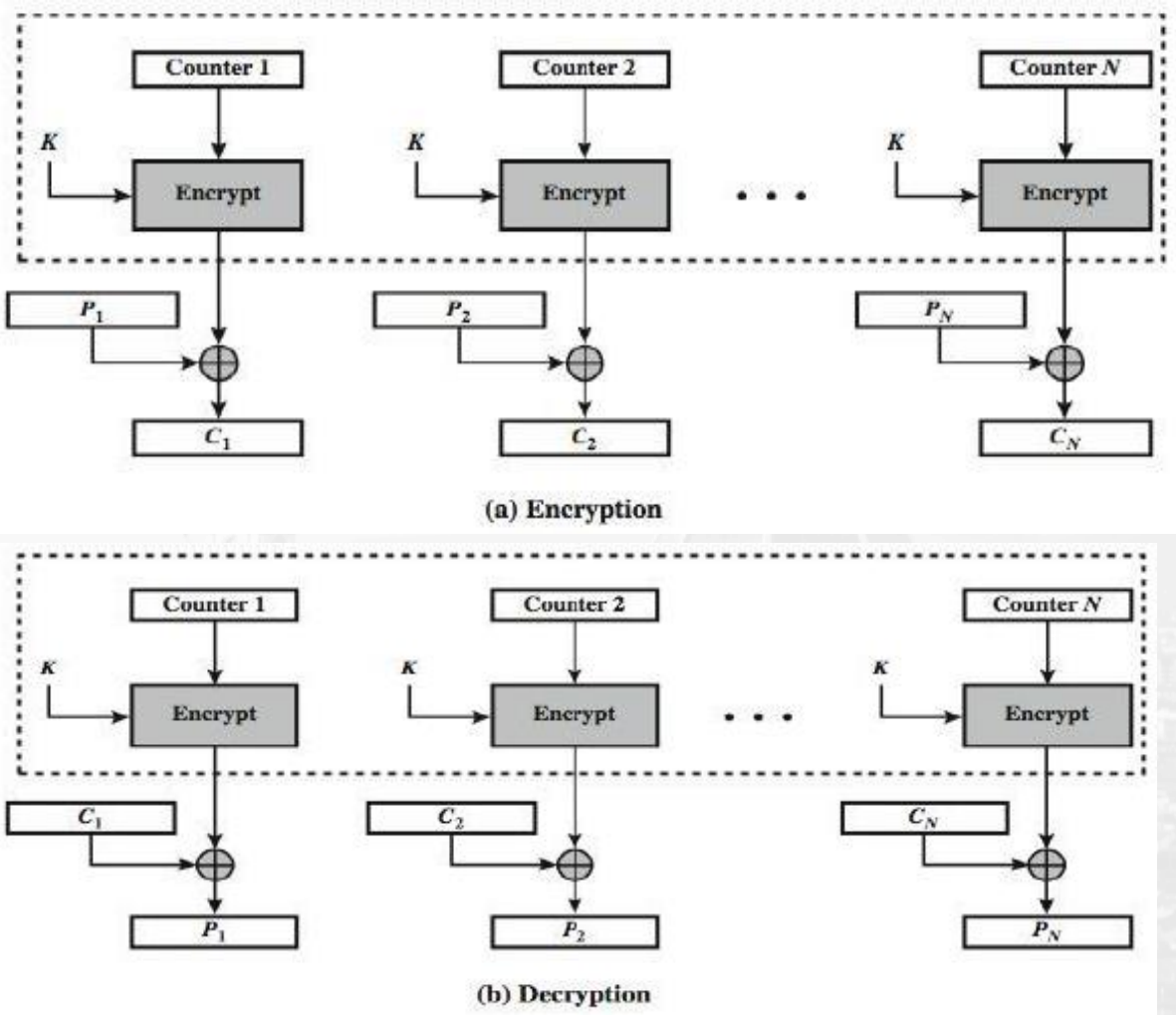
(a) Encryption



(b) Decryption

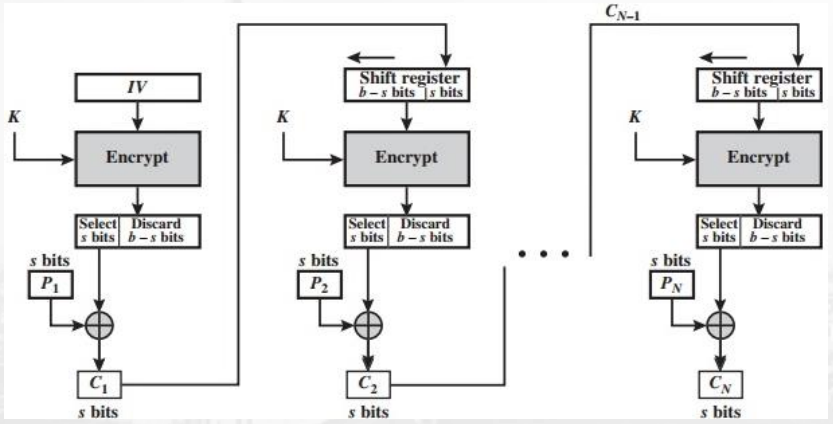
Counter (CTR) Mode

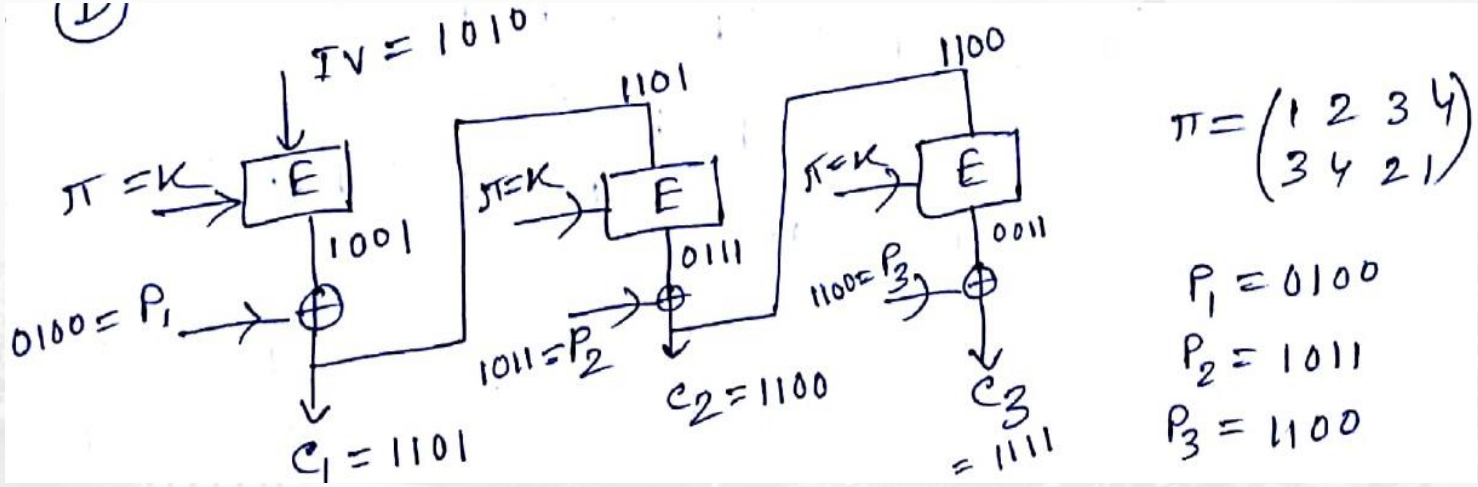
- It may be faster than of cipher block chaining mode.
- Encryption can be done in parallel.
- Padding is not required.
- Processing of plaintext blocks can be done randomly
- Integrity of the message is not maintained
- Reuse of counter value, compromise the security.
- It is ATM and IP sec



Problem 2

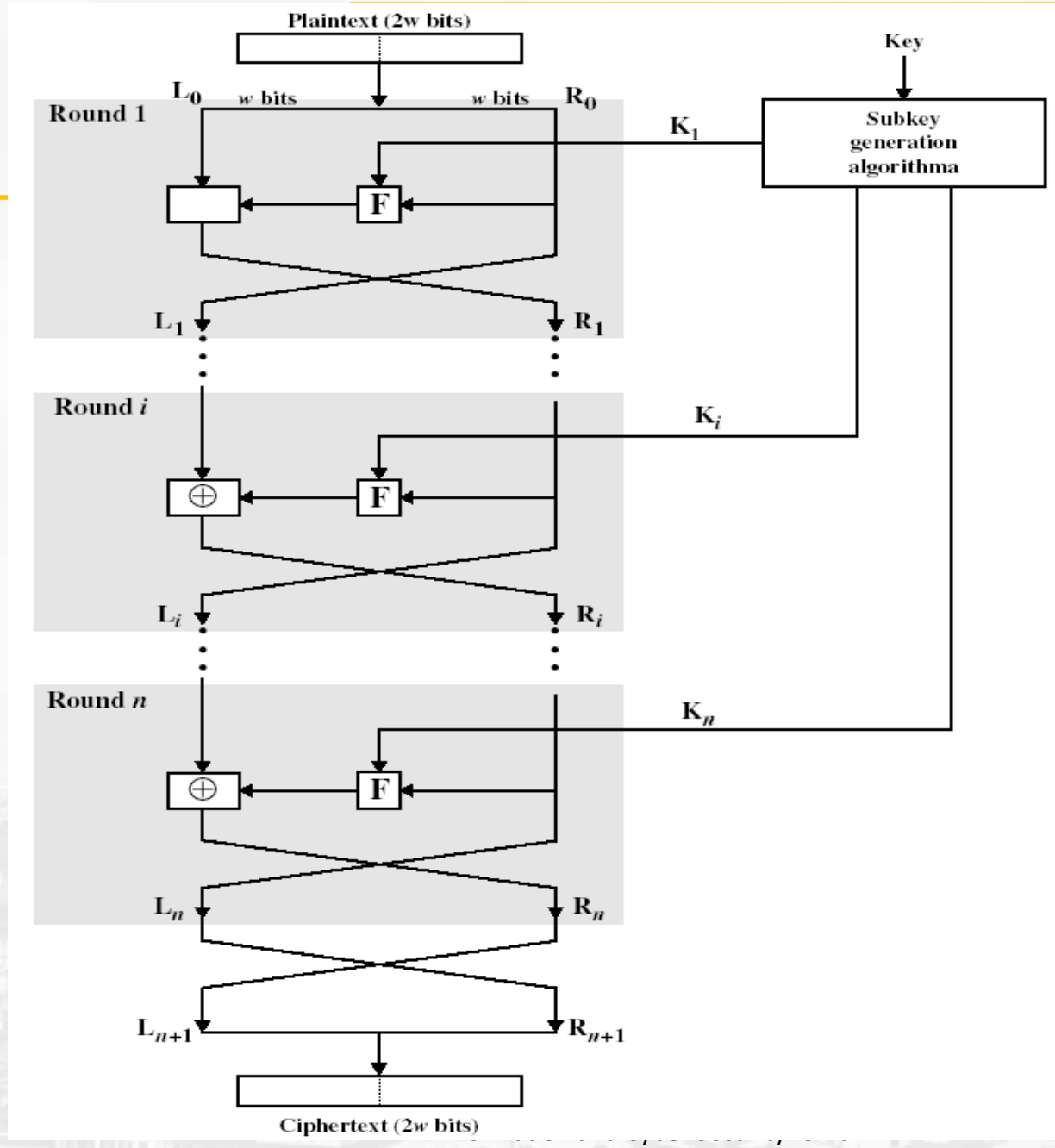
Consider the CFB of operation where the block cipher is permutation cipher and key is mutation $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$. If the initial vector is taken as 1010, the compute the ciphertext correspond to the plaintext 010010111100





Feistel Ciphers

- Most symmetric block ciphers are based on a **Feistel Cipher Structure**
- Block ciphers look like an extremely large substitution would need table of 2^{64} entries for a 64-bit block
- Horst Feistel devised the feistel cipher: based on concept of invertible product cipher
- partitions input block into two halves
  - process through multiple rounds which:
    - perform a substitution on left data half
    - based on round function of right half & sub key
    - then have permutation swapping halves
- The plaintext is divided into two halves (L_0 and R_0). Then the two halves pass through **n** rounds of processing then combine to produce the cipher block.
- Each round i has as input L_{i-1} and R_{i-1} derived from the previous round as well as a sub-key K_i derived from the overall K.


Plaintext (2w bits)
Key
L_0
w bits
w bits
R_0
Round 1
K_1
Subkey generation algorithma
F
L_1
R_1
Round i
K_i
F
L_i
R_i
Round n
K_n
F
L_n
R_n
L_{n+1}
R_{n+1}
Ciphertext (2w bits)

The design of Feistel cipher depends on following parameter:

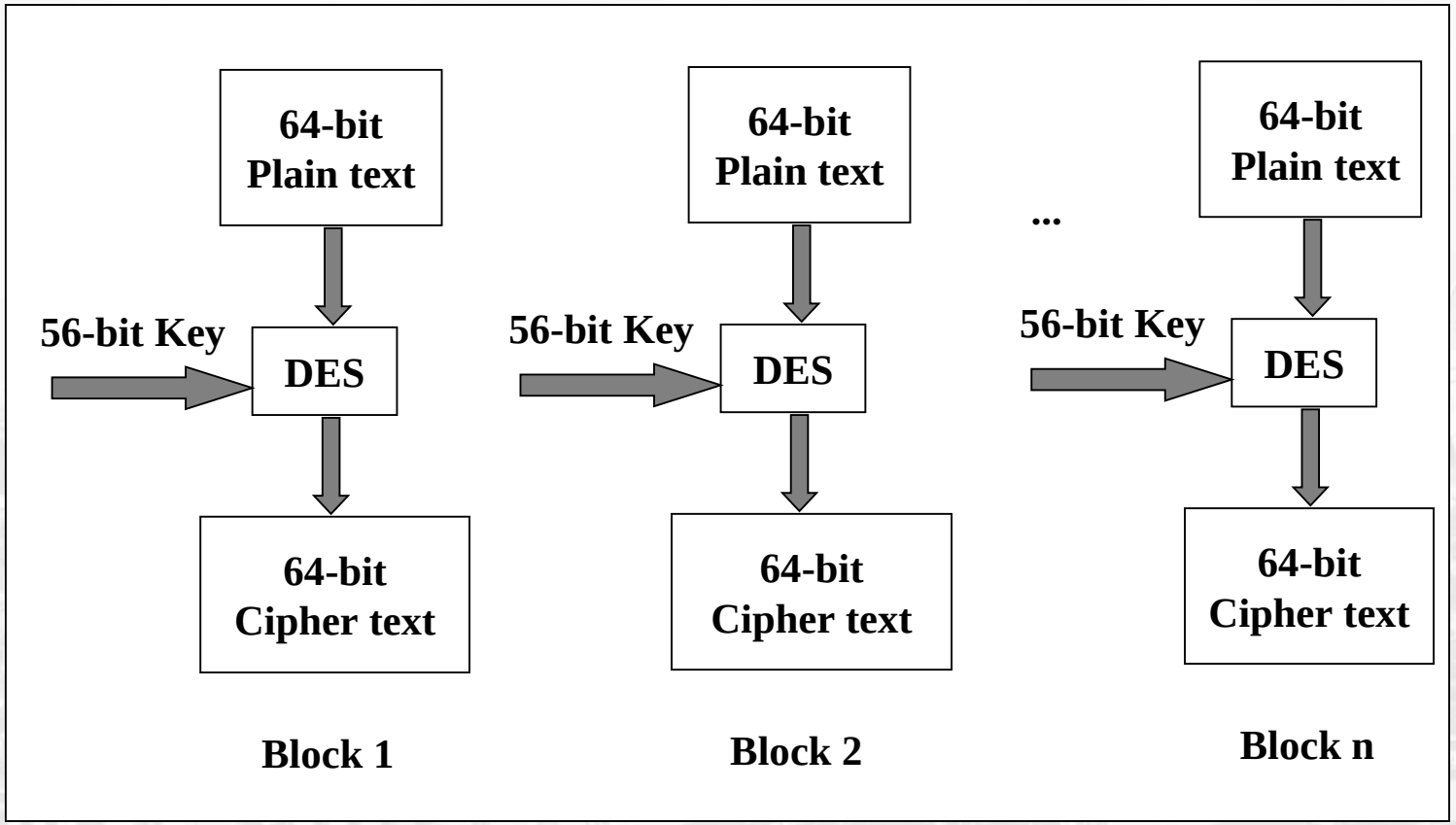
- **Block Size:** (larger block means greater security) 64 bits.
- **Key Size**: 56 - 128 bits.
- **Number of Rounds**: a single round offers inadequate security, a typical size is 16 rounds.
- **Sub-key Generation Algorithms**: greater complexity should lead to a greater difficulty of cryptanalysis.
- **Round function:** Again, greater complexity generally means greater resistance to cryptanalysis.

Data Encryption Standard (DES)

- IBM developed Lucifer cipher
  - by team led by Feistel
  - used 64-bit data blocks with 128-bit key
- then redeveloped as a commercial cipher with input from NSA and others in 1973 NBS issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES
- encrypts 64-bit data using 56-bit key
- DES has become widely used, especially in financial applications

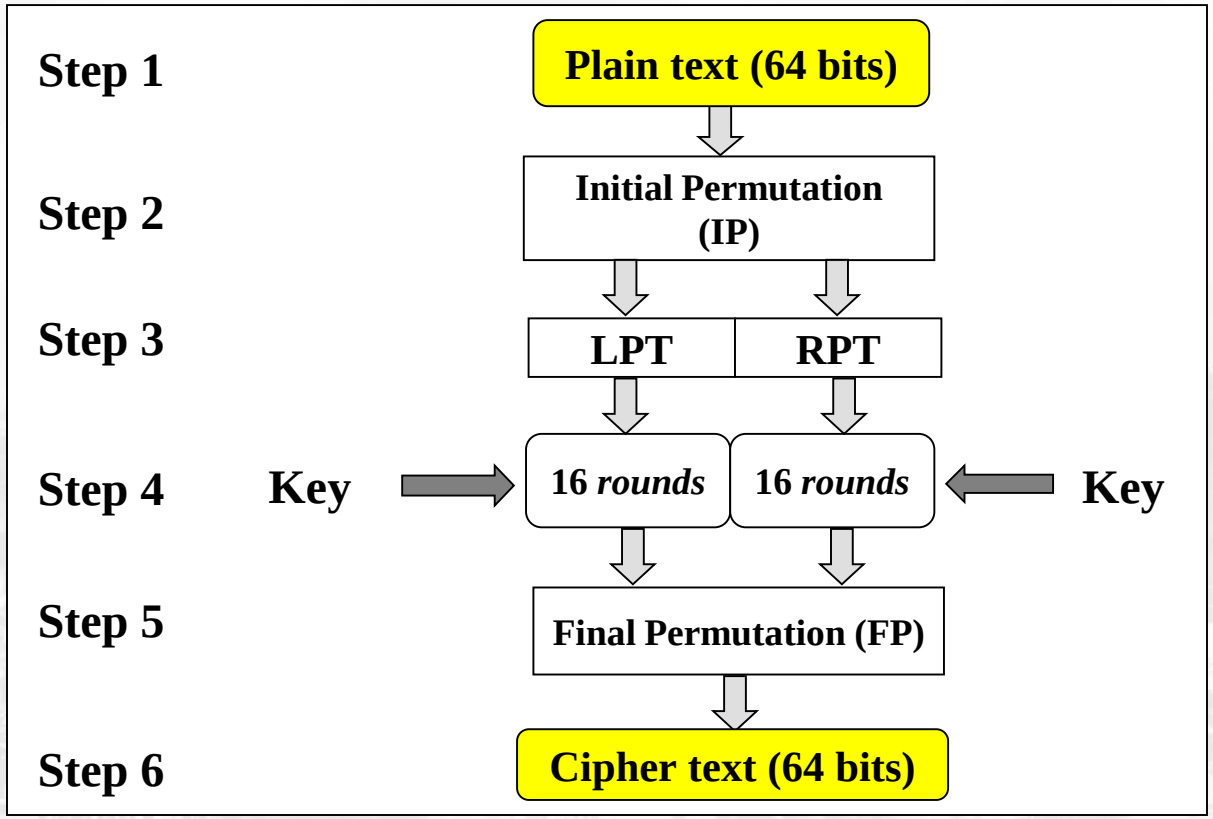
Conceptual View of DES

- Every 8th bit of the key is discarded to produce a 56-bit key
- Same algorithm and key are used for encryption and decryption



Broad Level Steps in DES

- DES is based on substitution (called as confusion) and transposition (called as diffusion)
- Each round performs the steps of substitution and transposition

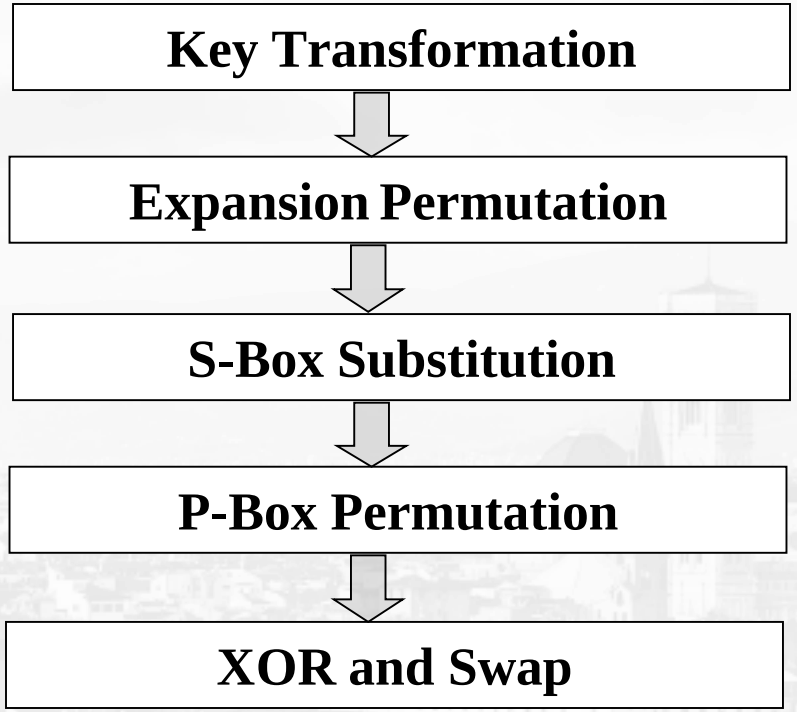


❖ Initial Permutation (IP)

- The first bit of the output is taken from the 58th bit of the input; the second bit from the 50th bit, and so on, with the last bit of the output taken from the 7th bit of the input. i.e. transposition

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

❖ Details of One Round in DES



Step 1: Key Transformation and Compression Permutation

- After the parity-bit drop, the key is divided into two 28-bit parts
- Each part is circularly shifted left by one or two bit
- The two parts are then combined to form a 56-bit part
- Choose 48 of the 56 bits

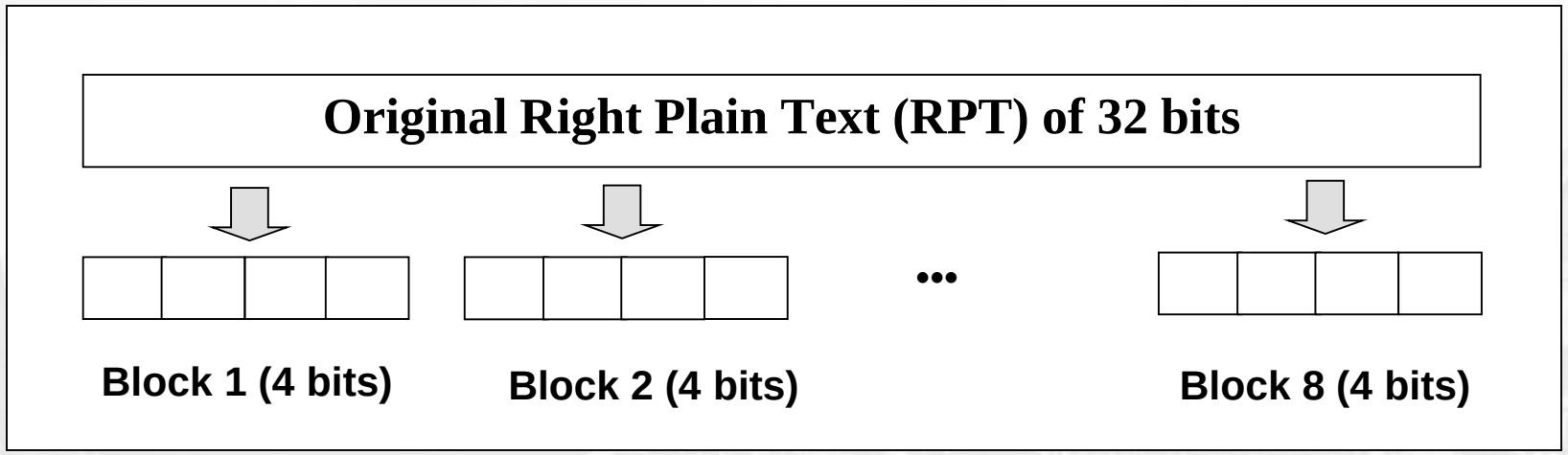
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

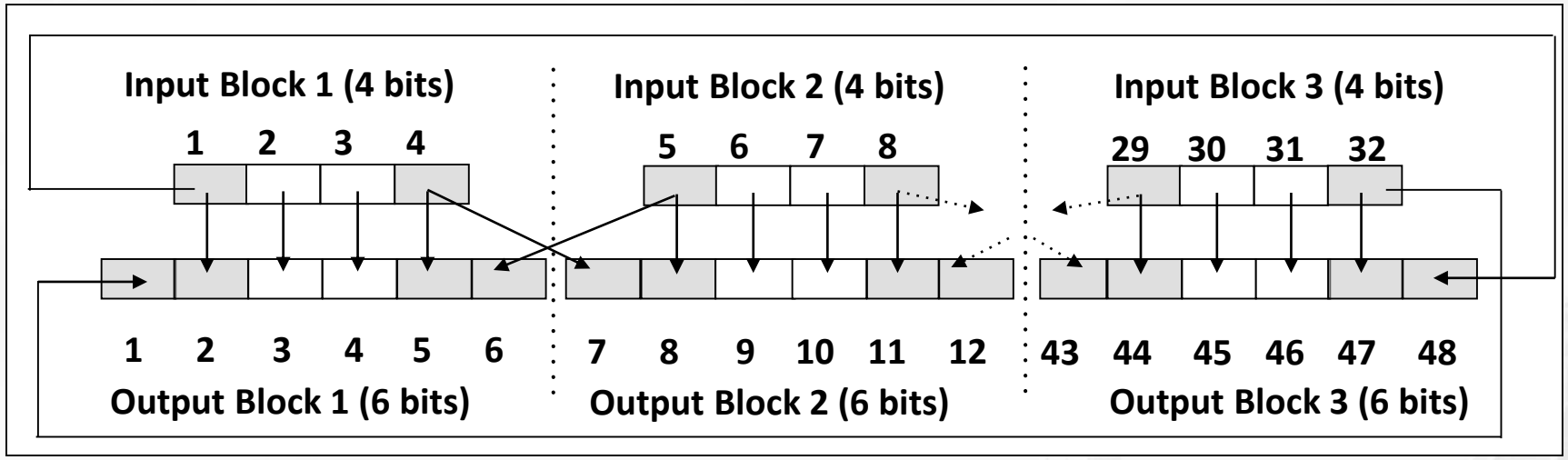
Compression Permutation

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Step 2: Expansion Permutation

- 32-bit RPT is divided into 8 blocks (each block 4-bits)
- Each 4-bit block is expanded to 6-bit block. Two bits -- repeated first and forth bits



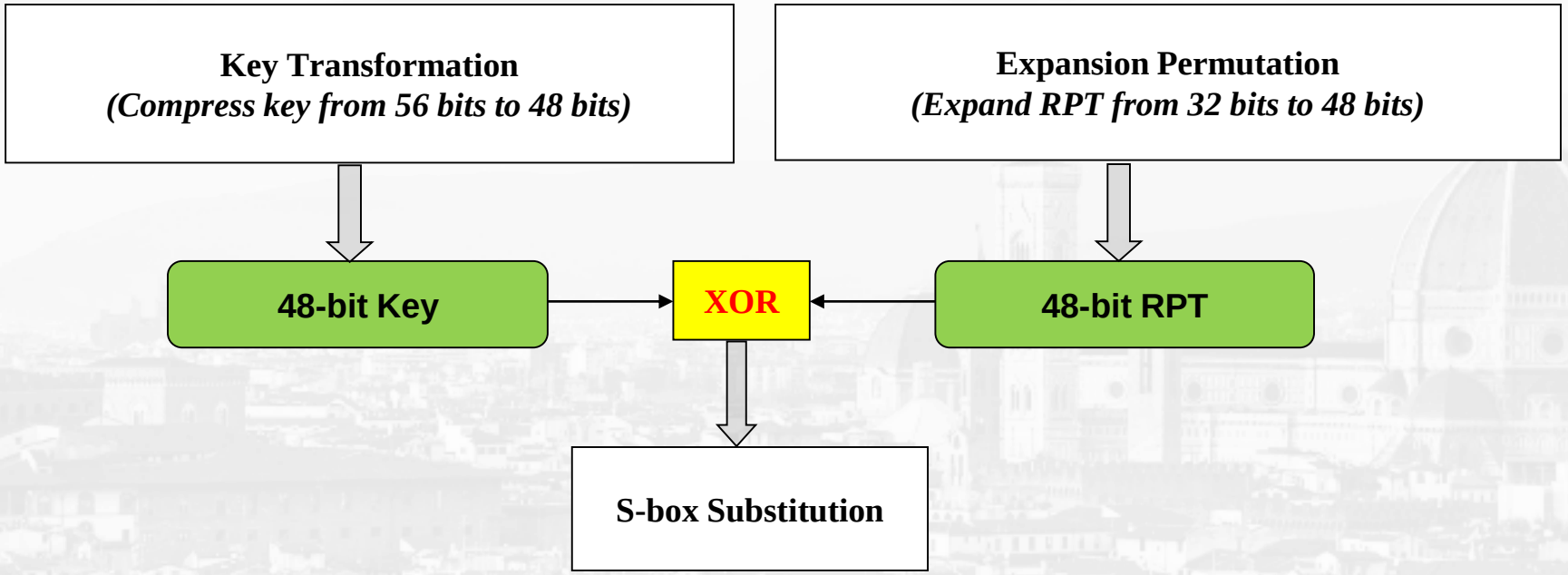


RPT Expansion Permutation Table

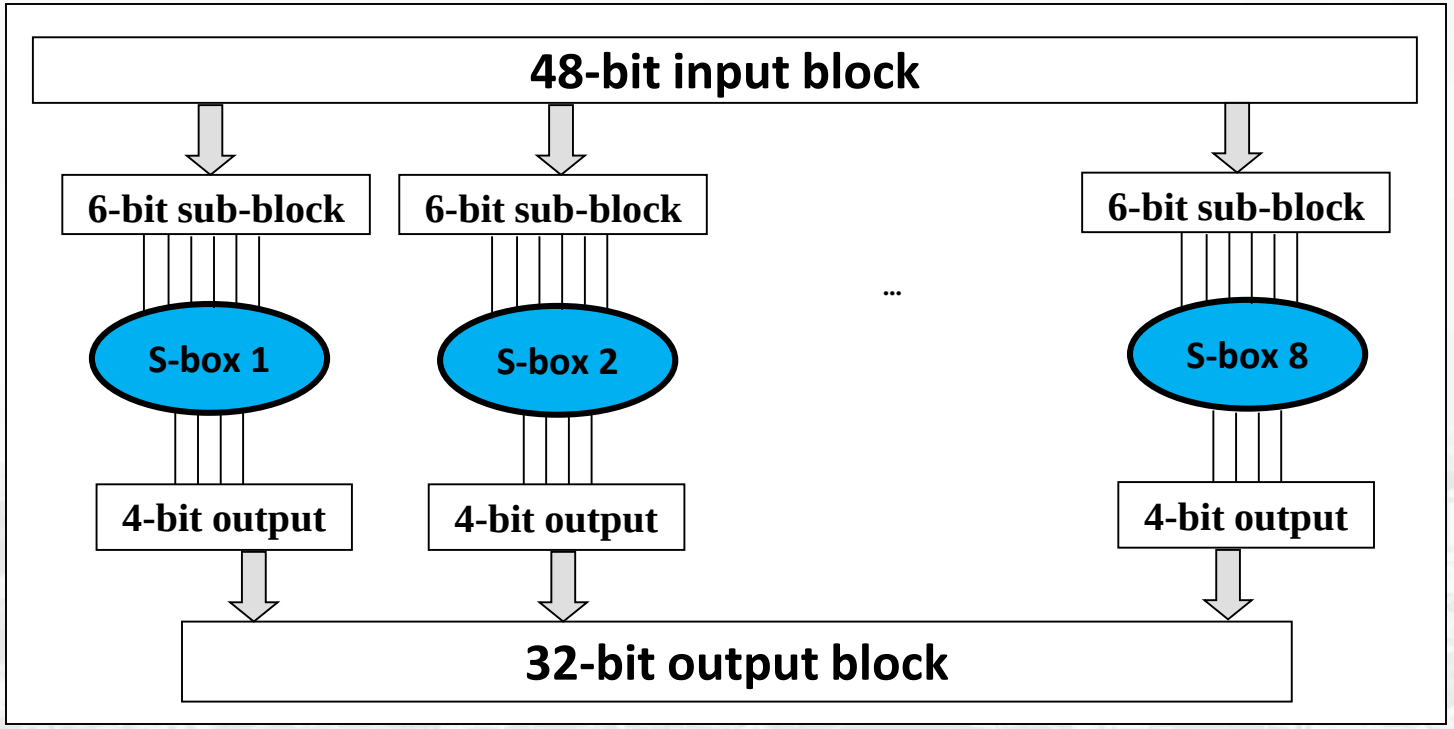
32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

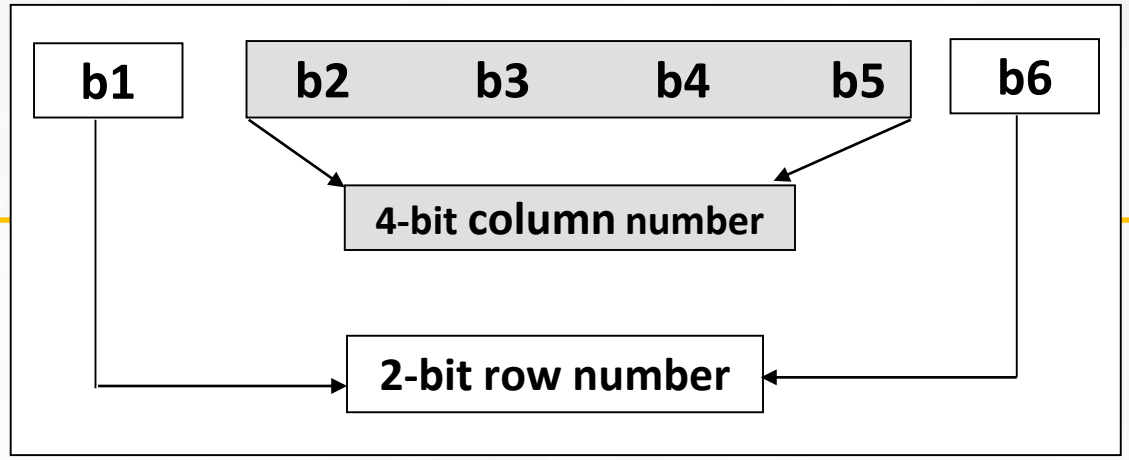
Step 3: S-box substitution

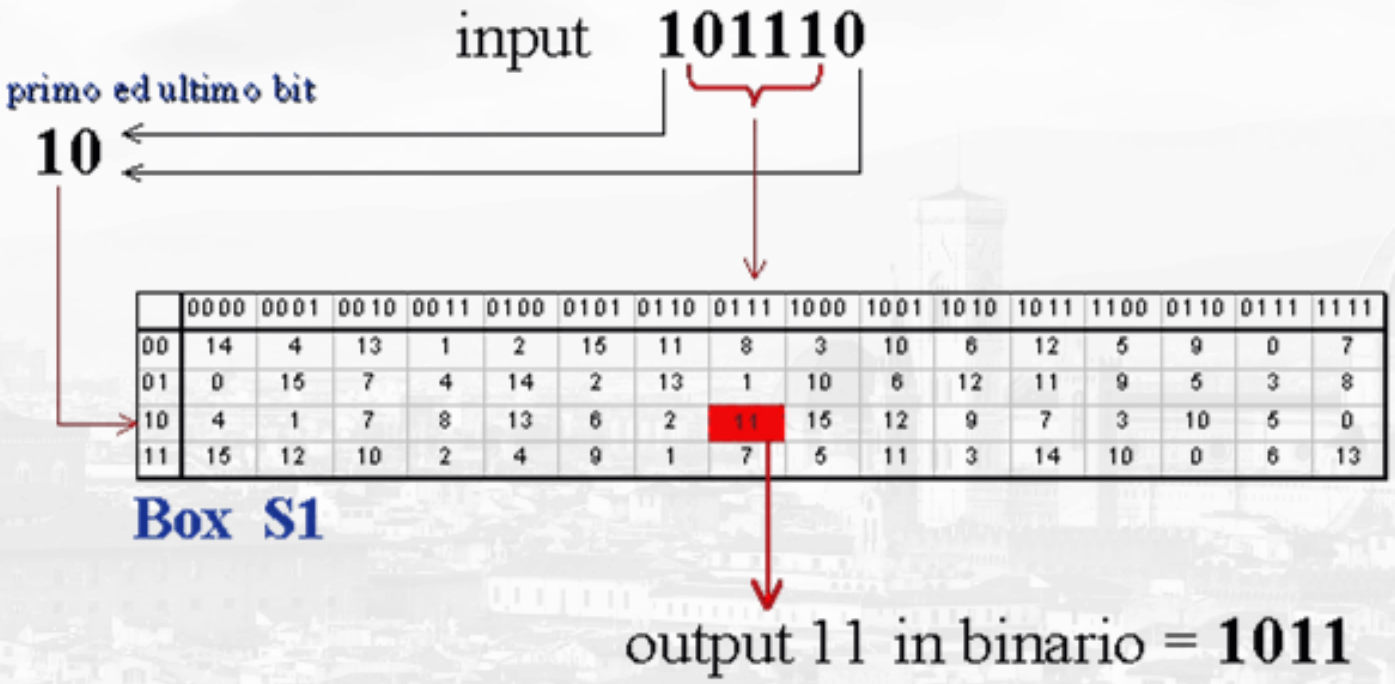
- Output: 32 bit



- Eight S-boxes that accept 6 bit inputs and produce 4 bit outputs







	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	0110	0111	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	7	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	10	2	4	9	1	7	5	11	3	14	10	0	6	13

S_1

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

S_2

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

S_3

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

S_4

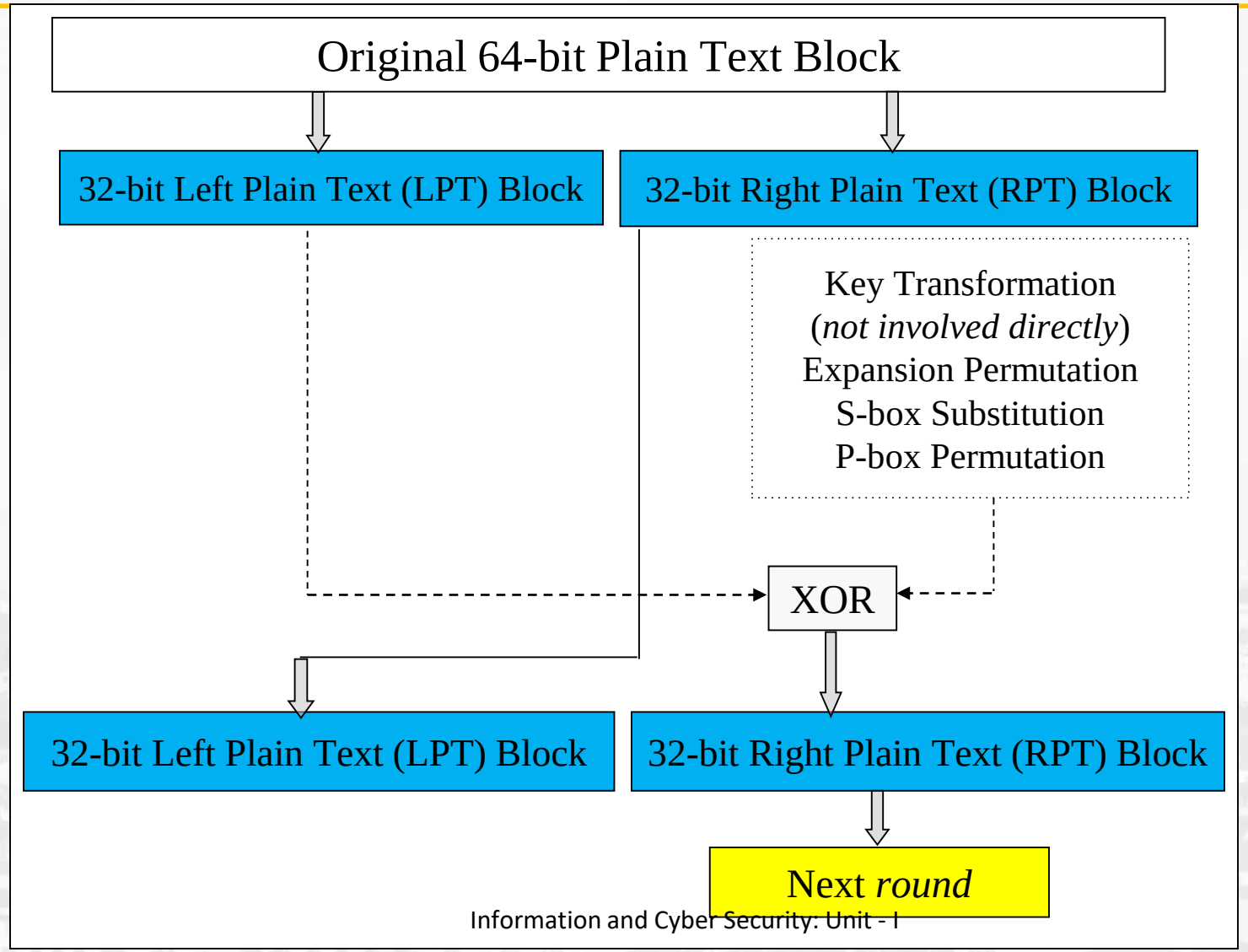
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Step 4: P-box permutation

- The output of S-box consists of 32 bits
- It is straight permutation. No bits are used twice and no bits are ignored
- Replacement of each bit with another bit

16	**7**	**20**	**21**	**29**	**12**	**28**	**17**	**1**	**15**	**23**	**26**	**5**	**18**	**31**	**10**
2	**8**	**24**	**14**	**32**	**27**	**3**	**9**	**19**	**13**	**30**	**6**	**22**	**11**	**4**	**25**

Step 5: XOR and Swap



❖ Final permutation

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

DES Decryption

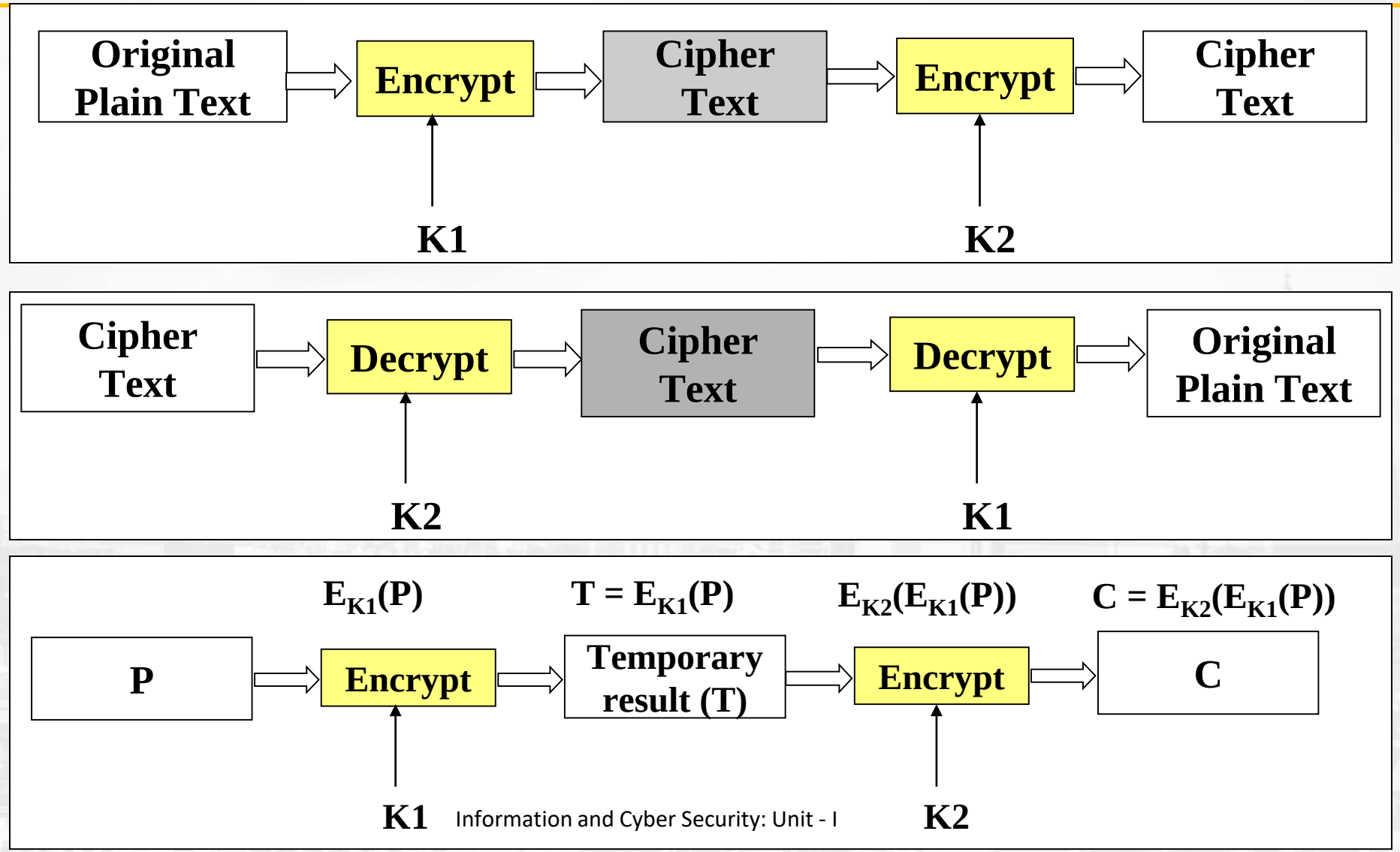
- Same algorithm and key are used for encryption and decryption
- Key reversal is used i.e. K16, K15, …… K1

Analysis of DES

- **Use of S-boxes**: The table used for substitution in DES are kept secret by IBM. It takes 17 years come up with internal design of the S-boxes.
- **Key Length**: There are 2^{56} possible keys i.e. 7.2 x 10^{16} keys. Thus, it seems that a brute-force attack on DES is impractical. A single computer performing one DES encryption per microsecond would require more than 1000 years to break DES.

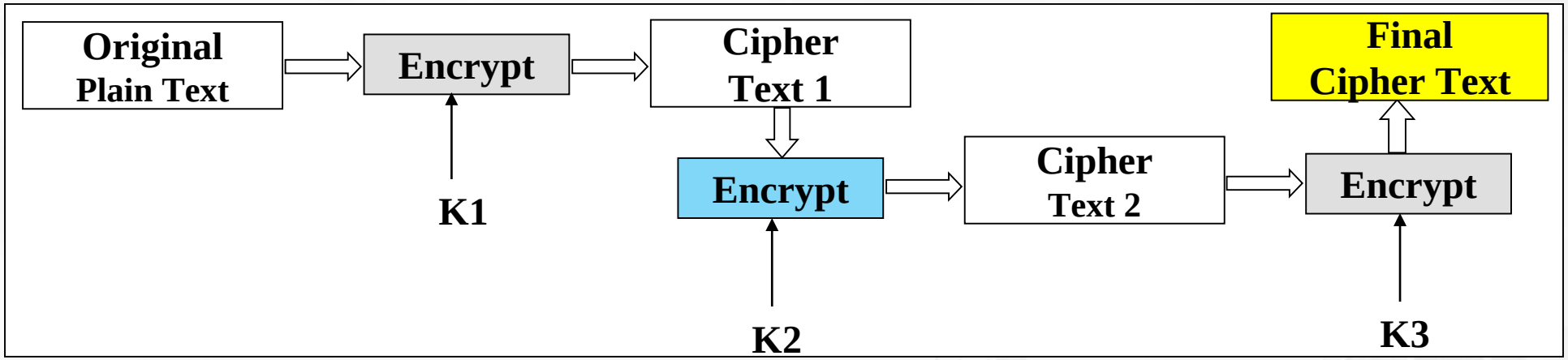
Variations of DES: Double DES

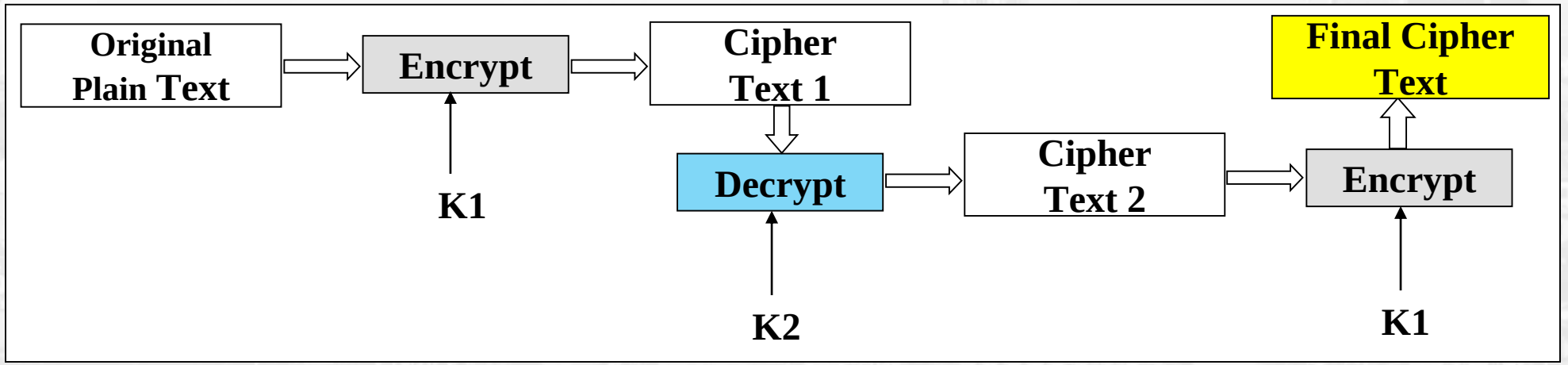
- Meet-in-the-middle attack



Triple DES

- Secure but more time for encryption





DES Weaknesses

Weaknesses in Cipher Design

S-boxes: At least three weaknesses are mentioned in the literature for S-boxes.

1. In S-box 4, the last three output bits can be derived in the same way as the first output bit by complementing some of the input bits.
2. Two specifically chosen inputs to an S-box array can create the same output.
3. It is possible to obtain the same output in a single round by changing bits in only three neighboring S-boxes.

P-boxes: One mystery and one weakness were found in the design of p-boxes:

1. It is not clear why the designers of DES used the initial and final permutations; these have no security benefits.
2. In the expansion permutation (inside the function), the first and fourth bits of every 4-bit series are repeated

Table 6.18 *Weak keys*

Keys before parities drop (64 bits)	*Actual key (56 bits)*
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

Table 6.19 *Semi-weak keys*

First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

Why A New Cipher?

- DES has the 56-bit key size and being too small.

- In January 1999, distributed.net and the Electronic Frontier Foundation collaborated to publicly break a DES key in 22 hours and 15 minutes

- **DES had outlived its usefulness:**
  - Vulnerabilities were becoming known
  - 56-bit key was too small
  - Too slow in software implementations

- **NIST wanted increased trust in cipher:**
  - Previous processes very closed
  - DES suspected of having 'back doors'

Advanced Encryption Standard (AES)

❖ **Background**

- On January 2, 1997, NIST announced the initiation of the AES development.

The point stipulated that:

- The algorithm must be a symmetric block cipher
- Key lengths of 128, 192, and 256 bits must be supported
- Block length: 128, 192, and 256 bits
- Both software and hardware implementations must be possible
- Possible implementation on smart-cards
- Royalty-free

The finalists and their scores were as follows: 15 Ciphers submitted

- Rijndael (from Joan Deamen and Vincent Rijmen, 86 votes).
- Serpent (from Ross Anderson, Eli Biham, and Lars Knudsen, 56 votes).
- Twofish (from a team headed by Bruce Schneier, 31 votes).
- RC6 (from RSA Laboratories, 23 votes).
- MARS (from IBM, 13 votes)

- In **November 2001**, Rijndael became a U. S. Government standard published as Federal Information Processing Standard FIPS 197.
- It is not a Feistel cipher.
  - It works in parallel over the whole input block.
  - Mode of operation: ECB

❖ The most powerful supercomputer in the world in 2017 was the Sunway TaihuLight in China. This beast is capable of a peak speed of 93.02 petaflops. This means that the most powerful computer in the world would still take some 885 quadrillion years to brute force a 128-bit AES key.

❖ The number of operations required to brute force a 256-bit cipher is 3.31 x 10^56. This is roughly equal to the number of atoms in the universe!

Rijndael's Encryption Algorithm

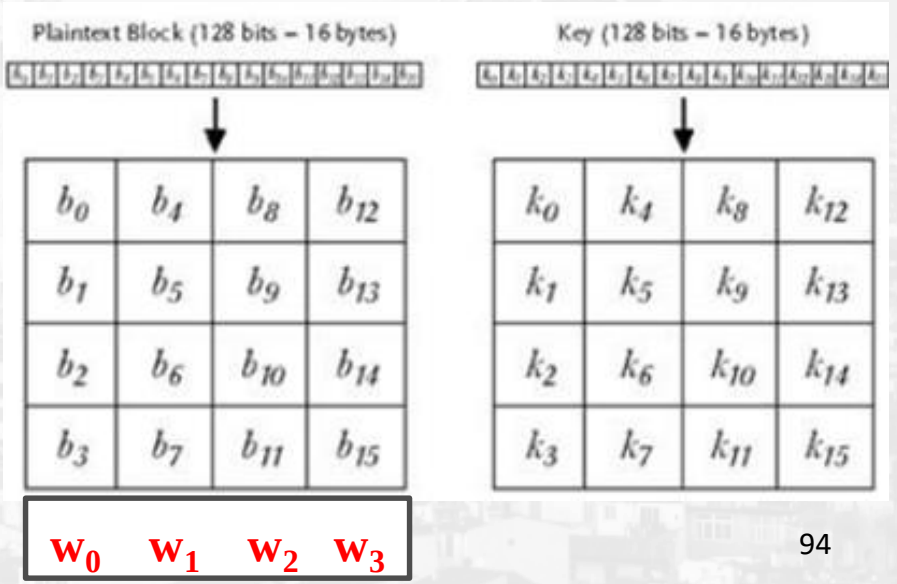
- The basic unit for processing in the AES algorithm is a byte.
- The AES algorithm's operations are performed on a two-dimensional array of bytes called the State. It is referred as $s_{r,c}$
- Block = 128 bits = 16 byte = $b_0\ b_1\ b_2 \ldots\ldots\ldots\ldots b_{15}$
- Key = 128 bits = 16 byte = $k_0\ k_1\ k_2 \ldots\ldots\ldots\ldots k_{15}$
- The four bytes in each column of the State array stand as one word

State representation

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

$w_0 = s_{0,0}\ s_{1,0}\ s_{2,0}\ s_{3,0}$ $\quad w_2 = s_{0,2}\ s_{1,2}\ s_{2,2}\ s_{3,2}$

$w_1 = s_{0,1}\ s_{1,1}\ s_{2,1}\ s_{3,1}$ $\quad w_3 = s_{0,3}\ s_{1,3}\ s_{2,3}\ s_{3,3}$

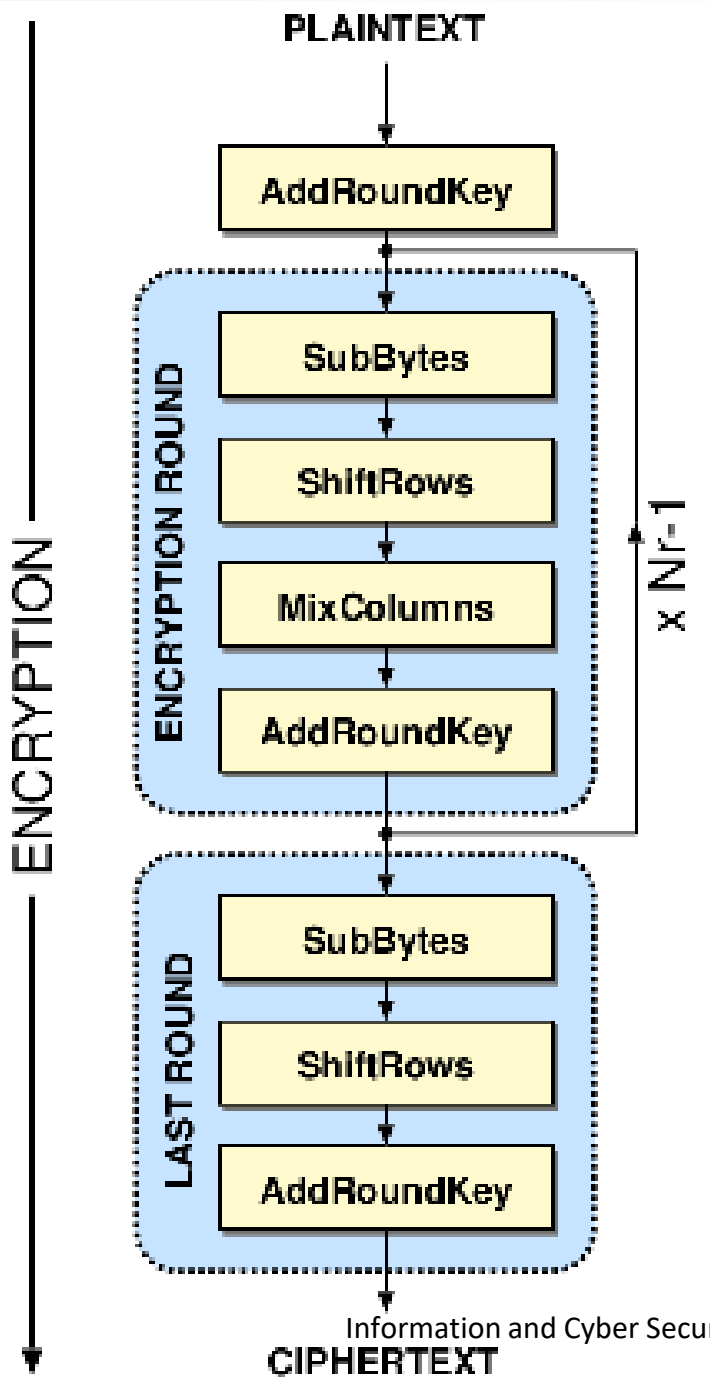


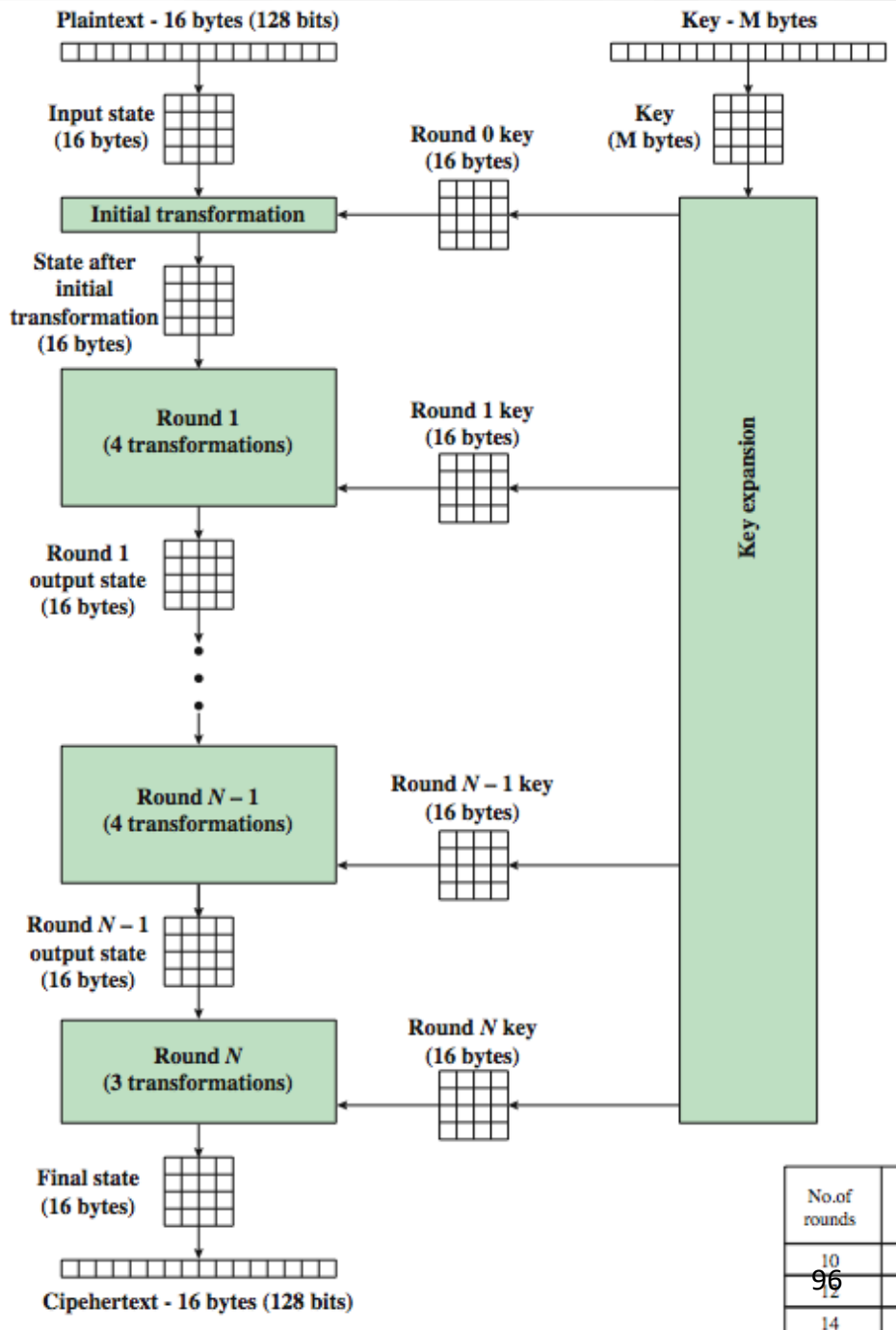
- Implemented as a 4 x 4 matrix, where each element in the matrix is one byte.
- Algorithm consists of an **initial round**, **Nr - 1** standard rounds where Nr is 10, 12, 14 depending on the block and key array sizes, and a **final round**.

Nr	Nb = 4	Nb = 6	Nb = 8
Nk = 4	10	12	14
Nk = 6	12	12	14
Nk = 8	14	14	14

- **Possible Round Operations**
  - ByteSub – Substitution of Bytes
  - Shift Row – Shifts Rows
  - MixColumn – Multiplies Columns
  - AddRoundKey – XORs by Key

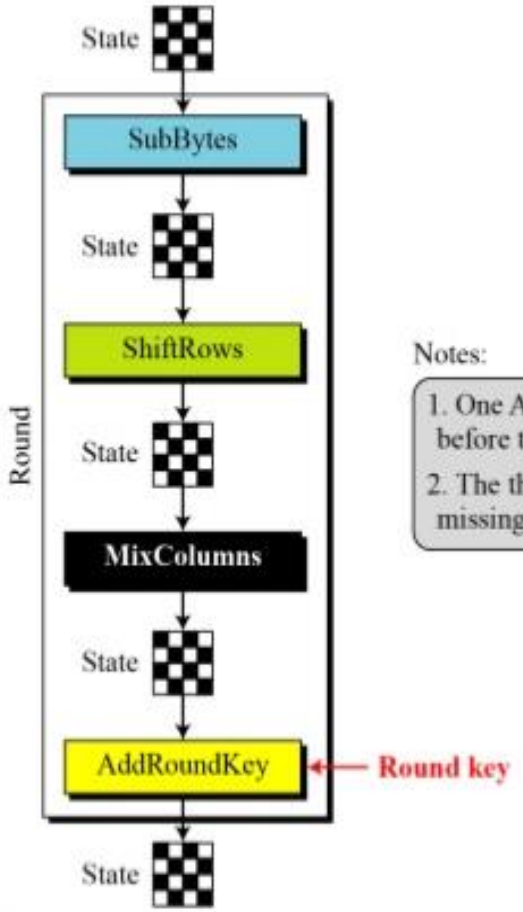
Encryption Algorithm





No.of rounds	Key Length (bytes)
10	16
12	24
14	32

Structure of Each Round



Notes:

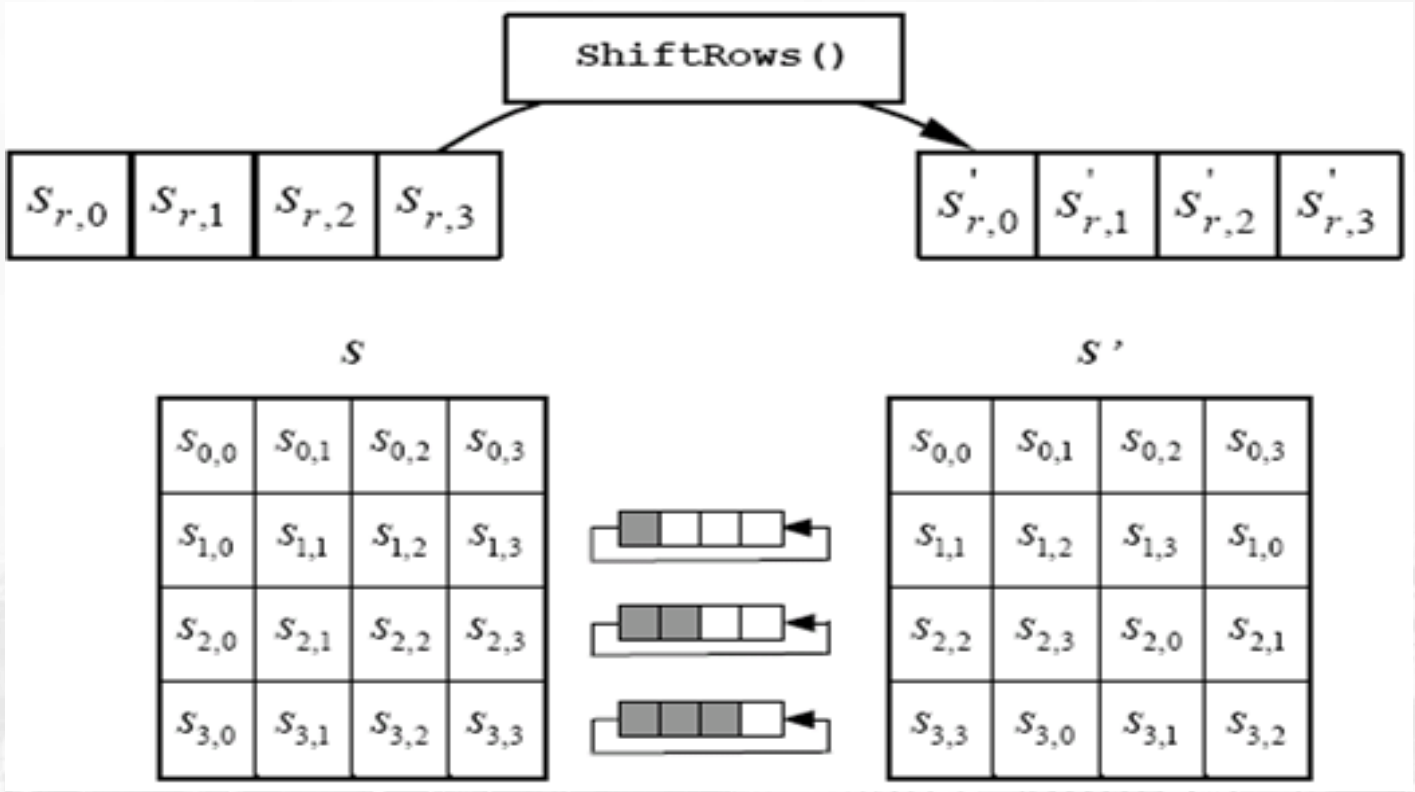
1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

SubBytes transformation

- 16 x 16 matrix whose entries are all distinct bytes.
- For example, if $s_{1,1} = \{53\}$, the result is {ed}.

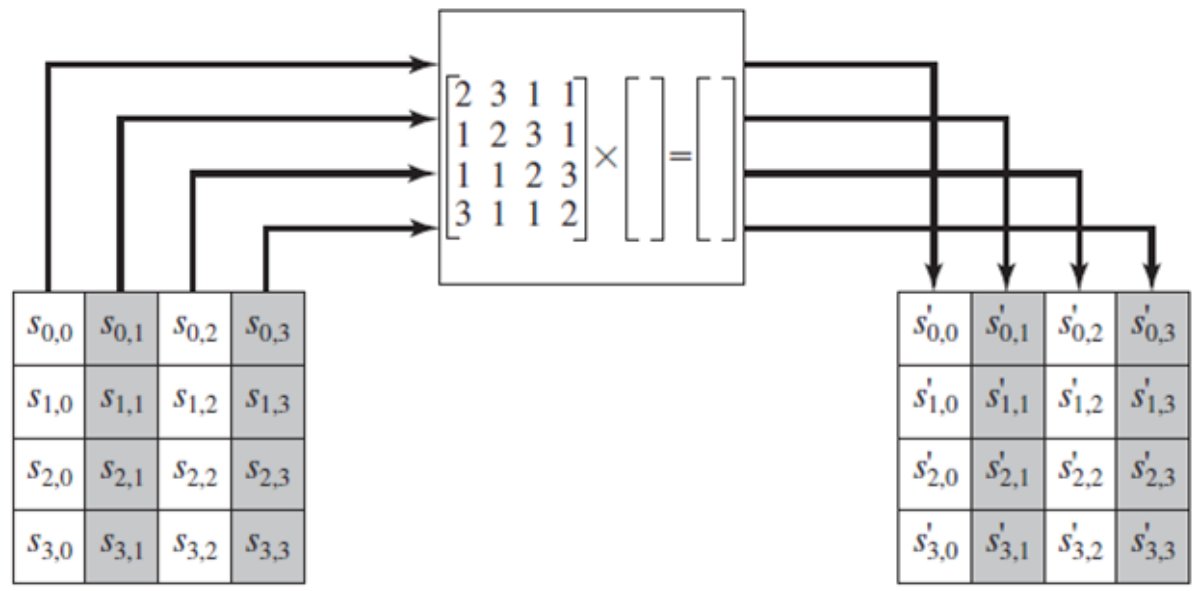
		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Shift-Rows transformation



MixColumn transformation

- The mixcolumn transformation operates on the state column-by-column.



$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} d4 & e0 & b8 & 1e \\ bf & 34 & 41 & 27 \\ 5d & 52 & 11 & 98 \\ 30 & ae & f1 & e5 \end{pmatrix}$$

02. {d4} $\oplus$ 03.{bf} $\oplus$ 01.{5d} $\oplus$ 01.{30} = {04}

Multiplication

- A polynomial is irreducible if its only divisors are one and itself.
- For the AES algorithm, this irreducible polynomial is

$m(x) = x^8 + x^4 + x^3 + x + 1$ or {01}{1b} in hexadecimal notation

E.g. {57} • {83} = {c1}, because

$$(x^6+x^4+x^2+x+1)\,(x^7+x+1) = x^{13} + x^{11} + x^9 + x^8 + x^7 + x^7 + x^5 + x^3 + x^2 + x + x^6 + x^4 + x^2 + x + 1$$

$$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$$

$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$ **modulo** $x^8 + x^4 + x^3 + x + 1$

$= x^7 + x^6 + 1$

Add Round Key transformation

- In this operation, a Round Key is applied to the State by a simple bitwise XOR .

Key schedule

- This consists of two components: the Key Expansion and the Round Key Selection.
- The basic principle is the following:
- The total number of Round Key bits is equal to the block length multiplied by the number of rounds plus 1. (i.e. 128 x 11 = 1408, 1408/32 = 44)
- The Cipher Key is expanded into an Expanded Key.
- Round Keys are taken from this Expanded Key in the following way: the first Round Key consists of the first Nb words, the second one of the following Nb words, and so on.

Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

Key expansion

The key expansion function depends on the value of Nk.

For Nk ≤ 6, we have:

```
KeyExpansion (byte Key [4*Nk], word W [Nb*(Nr+1)])
{
    for (i = 0; i < Nk; i++)
    {
            W[i] = (Key [4*i], Key [4*i+1], Key [4*i+2], Key [4*i+3])
    }
    for (i = Nk; i < Nb * (Nr + 1); i++)
    {       temp = W [i - 1];
            if (i % Nk = = 0)
            {
                    temp = SubByte (RotByte (temp)) XOR  Rcon [i / Nk]
            }
            W[i] = W[i - Nk]  XOR  temp
    }
}
```

- The round constants are independent of Nk and defined by:

Rcon[i] = (RC[i], ‘00’, ‘00’, ‘00’)

with RC[1] representing an element in GF(2^8) with a value of $x^{(i-1)}$ so that:

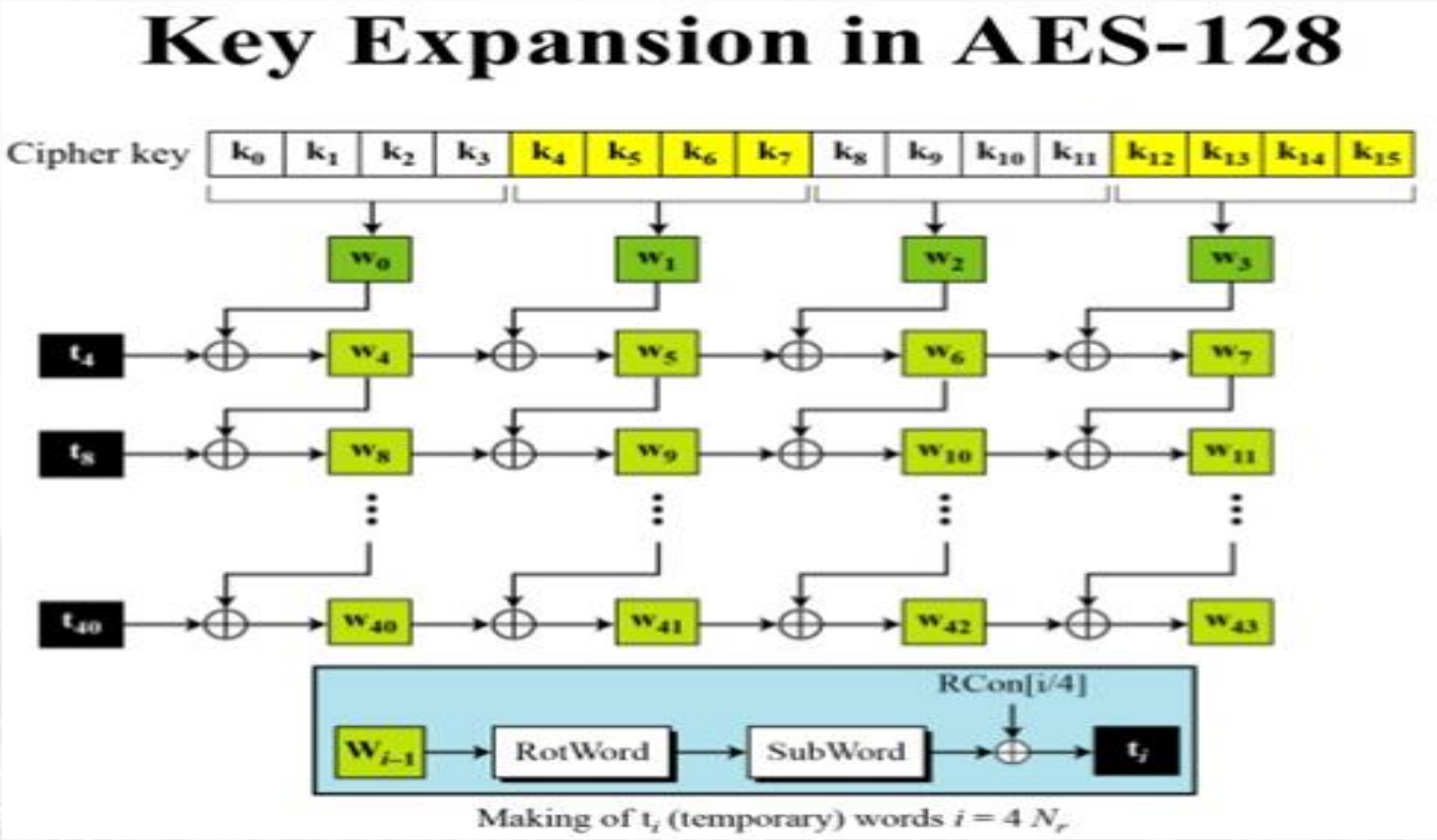
RC[1] = 1 (i.e. ‘01’)

RC[i] = x (i.e. ‘02’) • (RC[i-1]) = $x^{(i-1)}$

Table 7.4 *RCon constants*

Round	*Constant (RCon)*	*Round*	*Constant (RCon)*
1	$(\underline{\mathbf{01}}\ 00\ 00\ 00)_{16}$	6	$(\underline{\mathbf{20}}\ 00\ 00\ 00)_{16}$
2	$(\underline{\mathbf{02}}\ 00\ 00\ 00)_{16}$	7	$(\underline{\mathbf{40}}\ 00\ 00\ 00)_{16}$
3	$(\underline{\mathbf{04}}\ 00\ 00\ 00)_{16}$	8	$(\underline{\mathbf{80}}\ 00\ 00\ 00)_{16}$
4	$(\underline{\mathbf{08}}\ 00\ 00\ 00)_{16}$	9	$(\underline{\mathbf{1B}}\ 00\ 00\ 00)_{16}$
5	$(\underline{\mathbf{10}}\ 00\ 00\ 00)_{16}$	10	$(\underline{\mathbf{36}}\ 00\ 00\ 00)_{16}$

Key Expansion in AES-128



Making of t_i (temporary) words $i = 4\ N_r$

Key Expansion Examples

Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

For **Nk = 4**, which results in

w_0 = 2b7e1516 w_1 = 28aed2a6 w_2 = abf71588 w_3 = 09cf 4f3c

i (dec)	temp	After RotWord()	After SubWord()	Rcon[i/Nk]	After XOR with Rcon	w[i−Nk]	w[i] = temp XOR w[i−Nk]
4	09cf4f3c	cf4f3c09	8a84eb01	01000000	8b84eb01	2b7e1516	a0fafe17
5	a0fafe17					28aed2a6	88542cb1
6	88542cb1					abf71588	23a33939
7	23a33939					09cf4f3c	2a6c7605
8	2a6c7605	6c76052a	50386be5	02000000	52386be5	a0fafe17	f2c295f2
9	f2c295f2					88542cb1	7a96b943
10	7a96b943					23a33939	5935807a
11	5935807a					2a6c7605	7359f67f
12	7359f67f	59f67f73	cb42d28f	04000000	cf42d28f	f2c295f2	3d80477d
13	3d80477d					7a96b943	4716fe3e
14	4716fe3e					5935807a	1e237e44
15	1e237e44					7359f67f	6d7a883b
16	6d7a883b	7a883b6d	dac4e23c	08000000	d2c4e23c	3d80477d	ef44a541
17	ef44a541					4716fe3e	a8525b7f
18	a8525b7f					1e237e44	b671253b
19	b671253b					6d7a883b	db0bad00
20	db0bad00	0bad00db	2b9563b9	10000000	3b9563b9	ef44a541	d4d1c6f8
21	d4d1c6f8					a8525b7f	7c839d87
22	7c839d87					b671253b	caf2b8bc
23	caf2b8bc					db0bad00	11f915bc

Example: **Nb = 4 and Nk = 4**

Input : 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34

Key : 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

The Round Key values are taken from the Key Expansion example

Round Number	Start of Round	After SubByte	After ShiftRows	After MixColumns	Round Key Value
input	32 88 31 e0 43 5a 31 37 f6 30 98 07 a8 8d a2 34				⊕ 2b 28 ab 09 7e ae f7 cf 15 d2 15 4f 16 a6 88 3c =
1	19 a0 9a e9 3d f4 c6 f8 e3 e2 8d 48 be 2b 2a 08	d4 e0 b8 1e 27 bf b4 41 11 98 5d 52 ae f1 e5 30	d4 e0 b8 1e bf b4 41 27 5d 52 11 98 30 ae f1 e5	04 e0 48 28 66 cb f8 06 81 19 d3 26 e5 9a 7a 4c	⊕ a0 88 23 2a fa 54 a3 6c fe 2c 39 76 17 b1 39 05 =
2	a4 68 6b 02 9c 9f 5b 6a 7f 35 ea 50 f2 2b 43 49	49 45 7f 77 de db 39 02 d2 96 87 53 89 f1 1a 3b	49 45 7f 77 db 39 02 de 87 53 d2 96 3b 89 f1 1a	58 1b db 1b 4d 4b e7 6b ca 5a ca b0 f1 ac a8 e5	⊕ f2 7a 59 73 c2 96 35 59 95 b9 80 f6 f2 43 7a 7f =

3

aa	61	82	68
8f	dd	d2	32
5f	e3	4a	46
03	ef	d2	9a

ac	ef	13	45
73	c1	b5	23
cf	11	d6	5a
7b	df	b5	b8

ac	ef	13	45
c1	b5	23	73
d6	5a	cf	11
b8	7b	df	b5

75	20	53	bb
ec	0b	c0	25
09	63	cf	d0
93	33	7c	dc

⊕

3d	47	1e	6d
80	16	23	7a
47	fe	7e	88
7d	3e	44	3b

=

4

48	67	4d	d6
6c	1d	e3	5f
4e	9d	b1	58
ee	0d	38	e7

52	85	e3	f6
50	a4	11	cf
2f	5e	c8	6a
28	d7	07	94

52	85	e3	f6
a4	11	cf	50
c8	6a	2f	5e
94	28	d7	07

0f	60	6f	5e
d6	31	c0	b3
da	38	10	13
a9	bf	6b	01

⊕

ef	a8	b6	db
44	52	71	0b
a5	5b	25	ad
41	7f	3b	00

=

5

e0	c8	d9	85
92	63	b1	b8
7f	63	35	be
e8	c0	50	01

e1	e8	35	97
4f	fb	c8	6c
d2	fb	96	ae
9b	ba	53	7c

e1	e8	35	97
fb	c8	6c	4f
96	ae	d2	fb
7c	9b	ba	53

25	bd	b6	4c
d1	11	3a	4c
a9	d1	33	c0
ad	68	8e	b0

⊕

d4	7c	ca	11
d1	83	f2	f9
c6	9d	b8	15
f8	87	bc	bc

=

6

f1	c1	7c	5d
00	92	c8	b5
6f	4c	8b	d5
55	ef	32	0c

a1	78	10	4c
63	4f	e8	d5
a8	29	3d	03
fc	df	23	fe

a1	78	10	4c
4f	e8	d5	63
3d	03	a8	29
fe	fc	df	23

4b	2c	33	37
86	4a	9d	d2
8d	89	f4	18
6d	80	e8	d8

⊕

6d	11	db	ca
88	0b	f9	00
a3	3e	86	93
7a	fd	41	fd

=

7

26	3d	e8	fd
0e	41	64	d2
2e	b7	72	8b
17	7d	a9	25

f7	27	9b	54
ab	83	43	b5
31	a9	40	3d
f0	ff	d3	3f

f7	27	9b	54
83	43	b5	ab
40	3d	31	a9
3f	f0	ff	d3

14	46	27	34
15	16	46	2a
b5	15	56	d8
bf	ec	d7	43

⊕

4e	5f	84	4e
54	5f	a6	a6
f7	c9	4f	dc
0e	f3	b2	4f

=

8

5a	19	a3	7a
41	49	e0	8c
42	dc	19	04
b1	1f	65	0c

be	d4	0a	da
83	3b	e1	64
2c	86	d4	f2
c8	c0	4d	fe

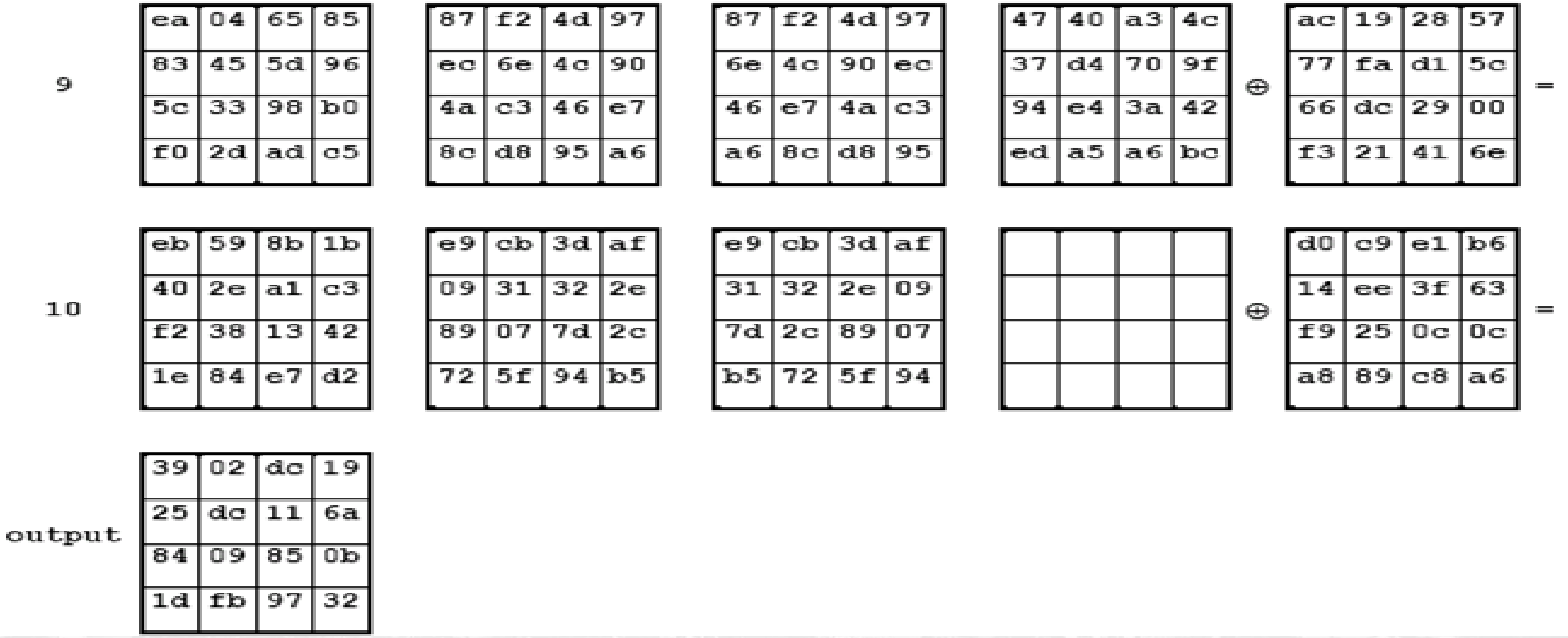
be	d4	0a	da
3b	e1	64	83
d4	f2	2c	86
fe	c8	c0	4d

00	b1	54	fa
51	c8	76	1b
2f	89	6d	99
d1	ff	cd	ea

⊕

ea	b5	31	7f
d2	8d	2b	8d
73	ba	f5	29
21	d2	60	2f

=


9
ea 04 65 85
83 45 5d 96
5c 33 98 b0
f0 2d ad c5
87 f2 4d 97
ec 6e 4c 90
4a c3 46 e7
8c d8 95 a6
87 f2 4d 97
6e 4c 90 ec
46 e7 4a c3
a6 8c d8 95
47 40 a3 4c
37 d4 70 9f
94 e4 3a 42
ed a5 a6 bc
⊕
ac 19 28 57
77 fa d1 5c
66 dc 29 00
f3 21 41 6e
=
10
eb 59 8b 1b
40 2e a1 c3
f2 38 13 42
1e 84 e7 d2
e9 cb 3d af
09 31 32 2e
89 07 7d 2c
72 5f 94 b5
e9 cb 3d af
31 32 2e 09
7d 2c 89 07
b5 72 5f 94
⊕
d0 c9 e1 b6
14 ee 3f 63
f9 25 0c 0c
a8 89 c8 a6
=
output
39 02 dc 19
25 dc 11 6a
84 09 85 0b
1d fb 97 32

- **Addition**
  - The addition is performed with the XOR operation (denoted by $\oplus$) - i.e., modulo 2 - so that

    $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, and $0 \oplus 0 = 0$.

$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$ (polynomial notation)

$\{01010111\} \oplus \{10000011\} = \{11010100\}$ (binary notation)

$\{57\} \oplus \{83\} = \{d4\}$ (hexadecimal notation)

Multiplication

- A polynomial is irreducible if its only divisors are one and itself.
- For the AES algorithm, this irreducible polynomial is

$m(x) = x^8 + x^4 + x^3 + x + 1$ or {01}{1b} in hexadecimal notation

E.g. {57} • {83} = {c1}, because

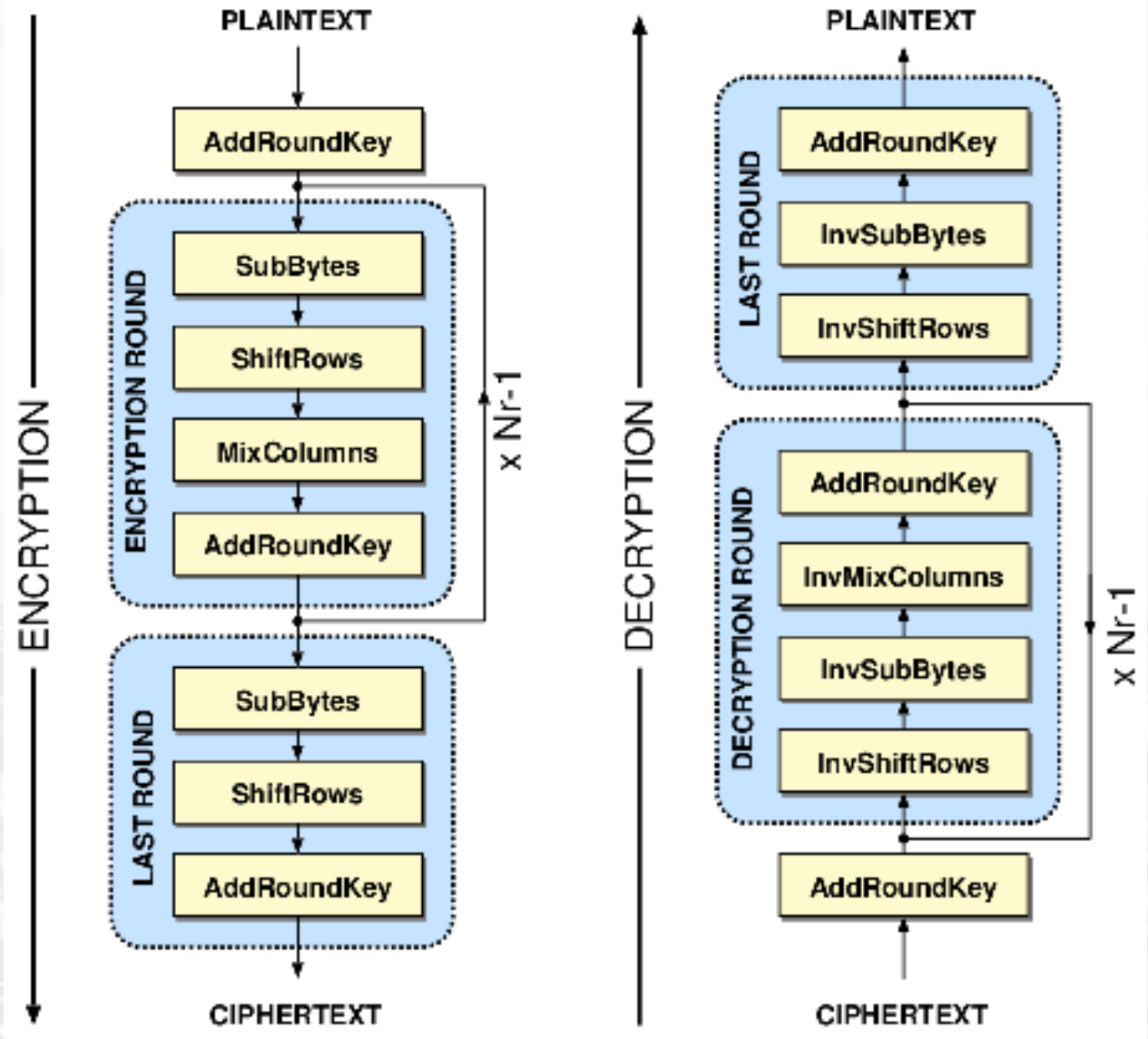
$$(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) = x^{13} + x^{11} + x^9 + x^8 + x^7 + x^7 + x^5 + x^3 + x^2 + x + x^6 + x^4 + x^2 + x + 1$$

$$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$$

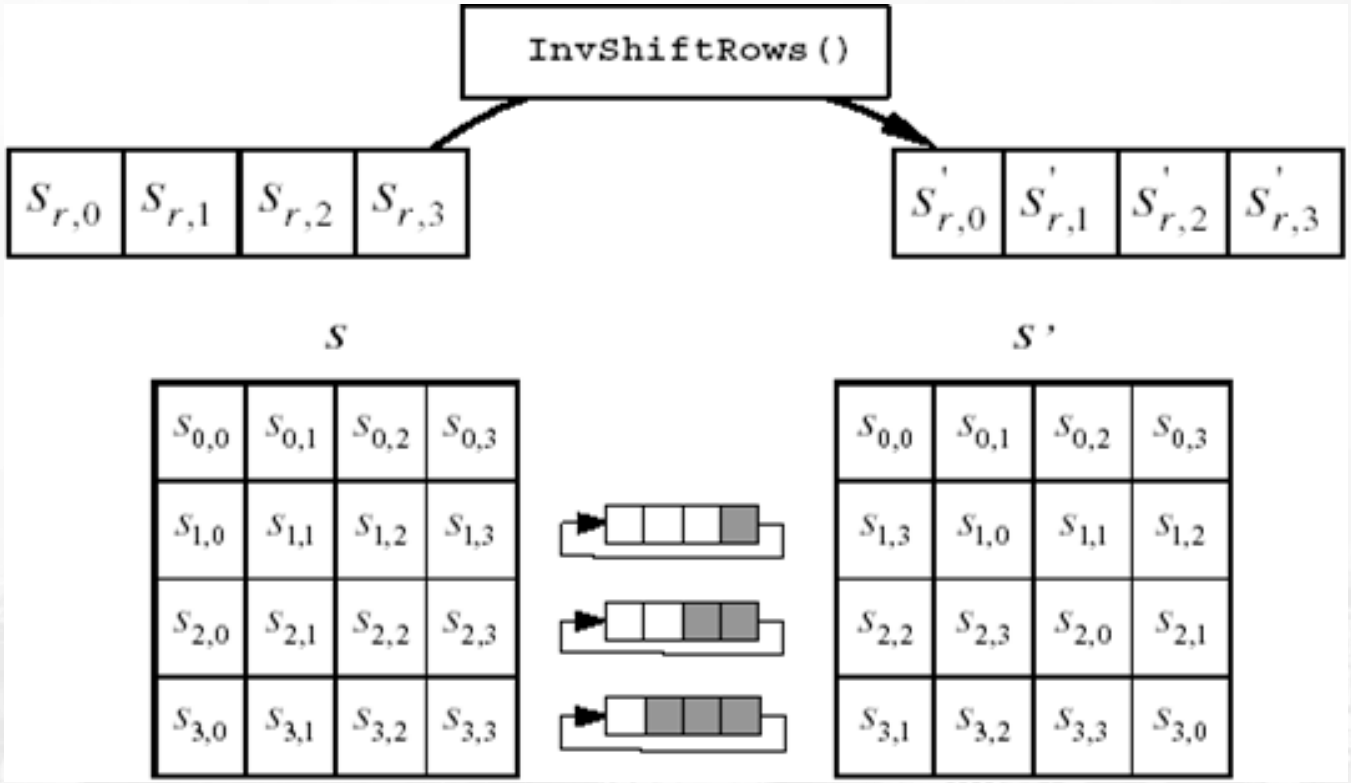
$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$ **modulo** $x^8 + x^4 + x^3 + x + 1$

$= x^7 + x^6 + 1$

❖ Rijndael's Decryption Algorithm



Inverse Shift-Rows transformation



MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Inverse Sub-Bytes transformation

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Inverse Mix-column transformation

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < Nb.$$

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$

$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$$

$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$$

Advantages

- Rijndael can be implemented to run at speeds unusually fast for a block cipher on a Pentium (Pro). There is a trade-off between table size/performance.
- Rijndael can be implemented on a Smart Card in a small amount of code, using a small amount of RAM and taking a small number of cycles. There is some ROM/performance trade-off.
- The round transformation is parallel by design, an important advantage in future processors and dedicated hardware.
- As the cipher does not make use of arithmetic operations, it has no bias towards big-or little endian processor architectures.

Limitations

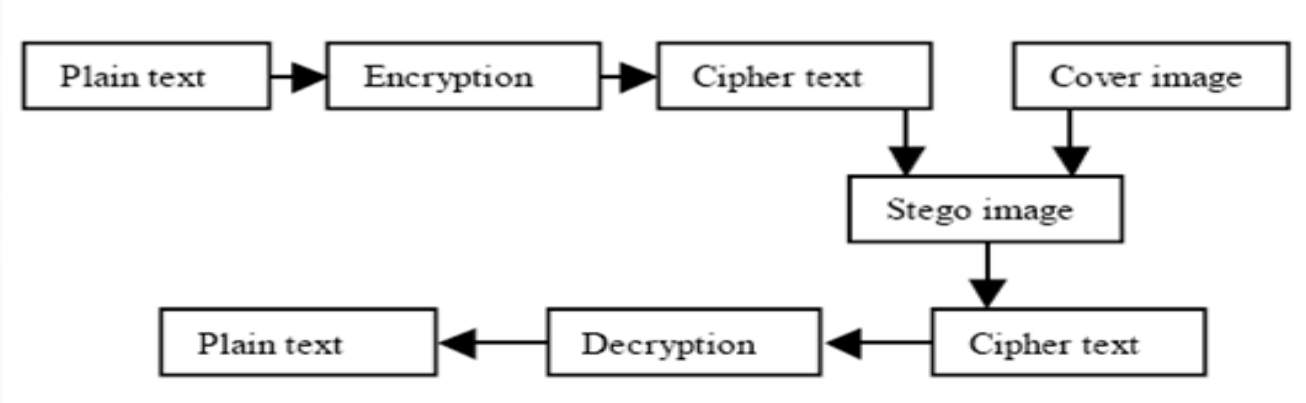
- The inverse cipher is less suited to be implemented on a smart card than the cipher itself: it takes more code and cycles. (Still, compared with other ciphers, even the inverse is very fast)
- In software, the cipher and its inverse make use of different code and/or tables.
- In hardware, the inverse cipher can only partially re-use the circuitry that implements the cipher.

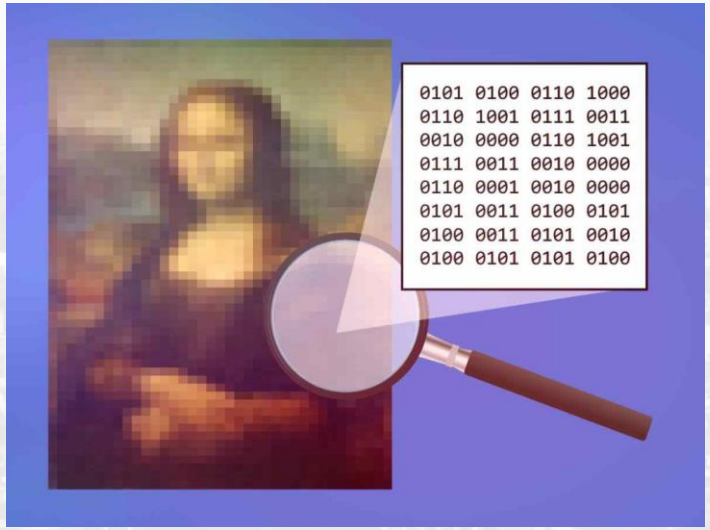
Comparison

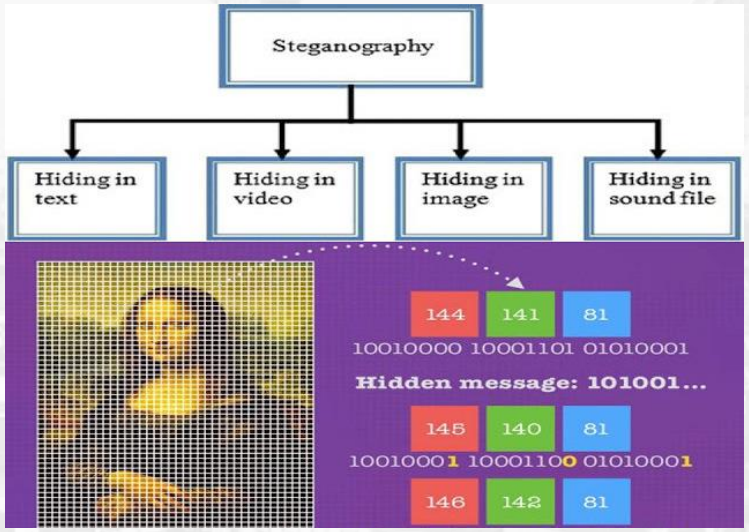
BASIS FOR COMPARISON	DES (DATA ENCRYPTION STANDARD)	AES (ADVANCED ENCRYPTION STANDARD)
Basic	In DES, the data block is divided into two halves.	In AES, the entire data block is processed as a single matrix.
Principle	DES work on Feistel Cipher structure.	AES work on block Cipher structure.
Plaintext	Plaintext is of 64 bits	Plaintext can be of 128,192, or 256 bits
Key size	DES in comparison to AES has smaller key size.	AES has larger key size as compared to DES.
Rounds	16 rounds	10 rounds for 128-bit algo, 12 rounds for 192-bit algo 14 rounds for 256-bit algo
Rounds Names	Expansion Permutation, Xor, S-box, P-box, Xor and Swap.	Subbytes, Shiftrows, Mix columns, Addroundkeys.
Security	DES has a smaller key which is less secure.	AES has large secret key comparatively hence, more secure.
Speed	DES is comparatively slower.	AES is faster.

Steganography

- Steganography is the art and science of writing hidden message in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message.
- Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, html or even floppy disks) with bits of different, invisible information.
- This hidden information can be plain text, cipher text or even images.
- In modern steganography, data is first encrypted by the usual means and then inserted, using a special algorithm, into redundant data that is part of a particular file format such as a JPEG image, Bitmap image.


Plain text
Encryption
Cipher text
Cover image
Stego image
Plain text
Decryption
Cipher text


0101 0100 0110 1000
0110 1001 0111 0011
0010 0000 0110 1001
0111 0011 0010 0000
0110 0001 0010 0000
0101 0011 0100 0101
0100 0011 0101 0010
0100 0101 0101 0100


Steganography
Hiding in text
Hiding in video
Hiding in image
Hiding in sound file
144
141
81
10010000 10001101 01010001
Hidden message: 101001...
145
140
81
10010001 10001100 01010001
146
142
81

❖ **Advantages:**

- No one suspects existence of message
- Highly secure

❖ **Disadvantages:**

- It requires a lot of overhead to hide a relatively few bits of information

Steganography	Cryptography
Unknown message passing	Known message passing
Steganography prevents discovery of the very existence of communication	Encryption prevents an unauthorized party from discovering the contents of a communication
Little known technology	Common technology
Technology still being develop for certain formats	Most of algorithm known by all
Once detected message is known	Strong current algorithm are resistant to attacks, larger expensive computing power is required for cracking
Steganography does not alter the structure of the secret message	Cryptography alter the structure of the secret message

Thank You !!!!!!