Fundamentos de Segurança Informática

LEI

2025/2026

T04 – Symmetric Encryption

T04 – Symmetric Encryption

Fundamental concepts

Two types of cryptographic systems

Asymmetric encryption systems:

- Different keys (of the same keypair) used to encrypt and decrypt
- Addresses the key distribution problem of symmetric systems (the public key may be distributed freely)

Symmetric encryption systems:

- Much faster than asymmetric systems
- Same key used to encrypt and decrypt: key distribution problem

Note: the two types of encryption are **complementary** and useful in the context of Internet security protocols and solutions!

Definitions

Plaintext
- An original message

Ciphertext
- The coded message

Enciphering/encryption
- The process of converting from plaintext to ciphertext

Deciphering/decryption
- Restoring the plaintext from the ciphertext

Cryptography
- The area of study of the many schemes used for encryption

Cryptographic system/cipher
- A scheme

Cryptanalysis
- Techniques used for deciphering a message without any knowledge of the enciphering details

Cryptology
- The areas of cryptography and cryptanalysis

Symmetric Cipher Model

- Symmetric (or conventional) encryption was the only type of encryption in use prior to the development of public-key encryption in the 1970s
- Remains the **most widely used** of the two types of encryption

Requirements for secure use of conventional encryption:

✓A **strong encryption algorithm**: an opponent should not be able to decrypt ciphertext or discover the key even in possession of a number of ciphertexts, together with the corresponding plaintext

✓Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure (**out-of-band** mechanisms)

Simplified Model of Symmetric Encryption

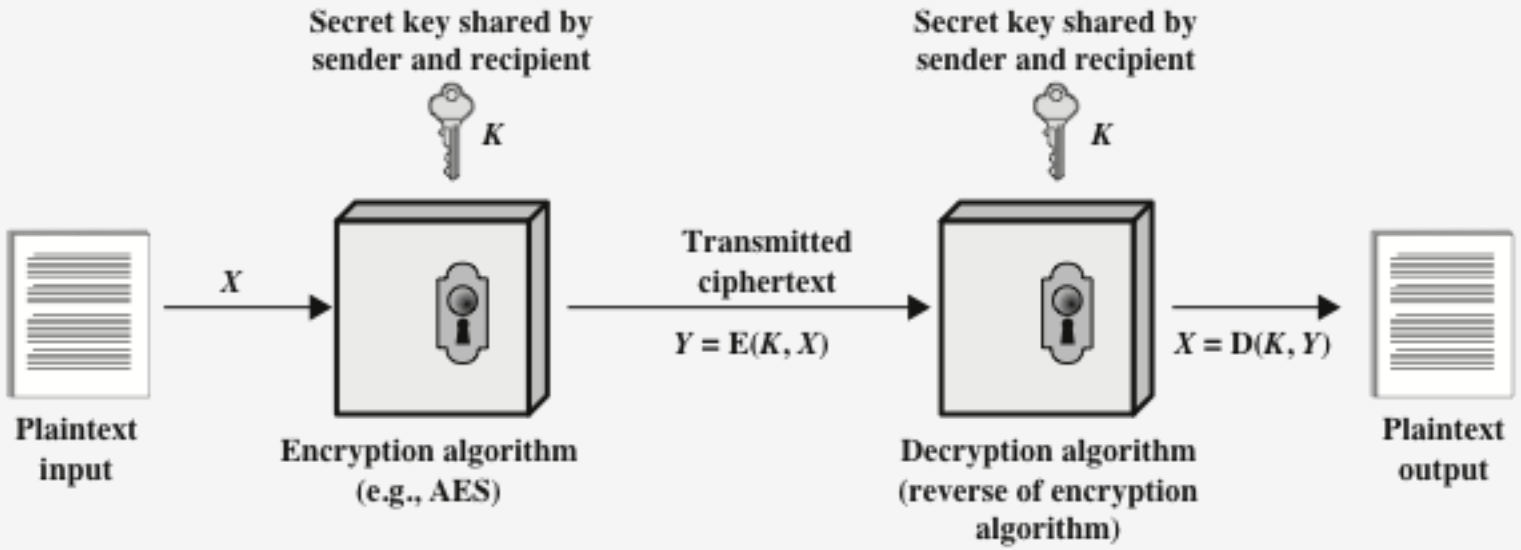


Figure 3.1 Simplified Model of Symmetric Encryption

Model of Symmetric Cryptosystem

$X = [X_1, X_2, \ldots, X_m]$

$K = [K_1, K_2, \ldots, K_m]$

$Y = [Y_1, Y_2, \ldots, Y_m]$

$Y = E(K, X)$

$X = D(K, Y)$

- We assume that an opponent knowns E and D
- He or she may be interested only in a particular message (estimating X) or in reading future messages as well (recover K)

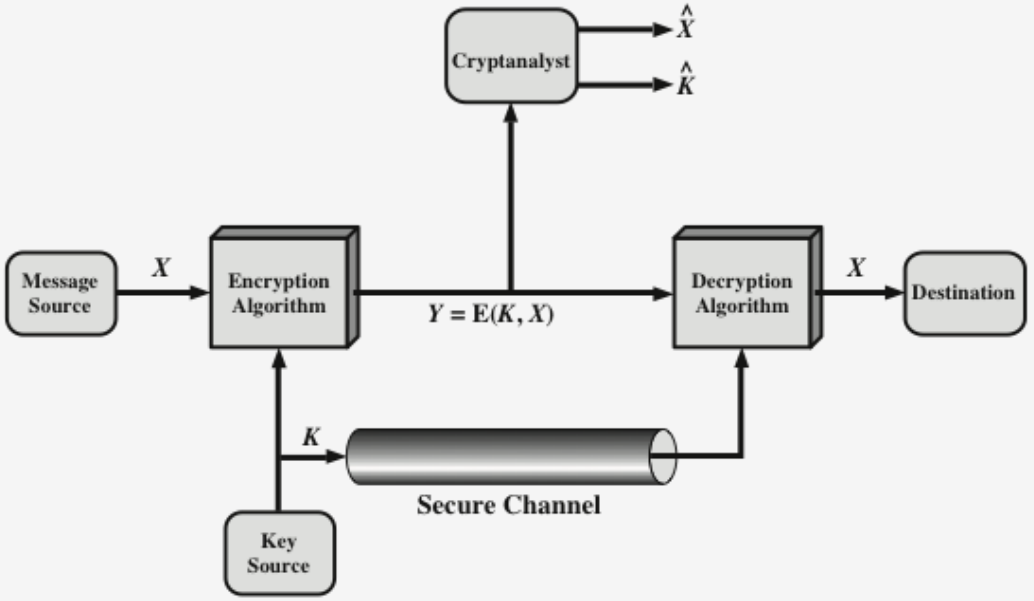
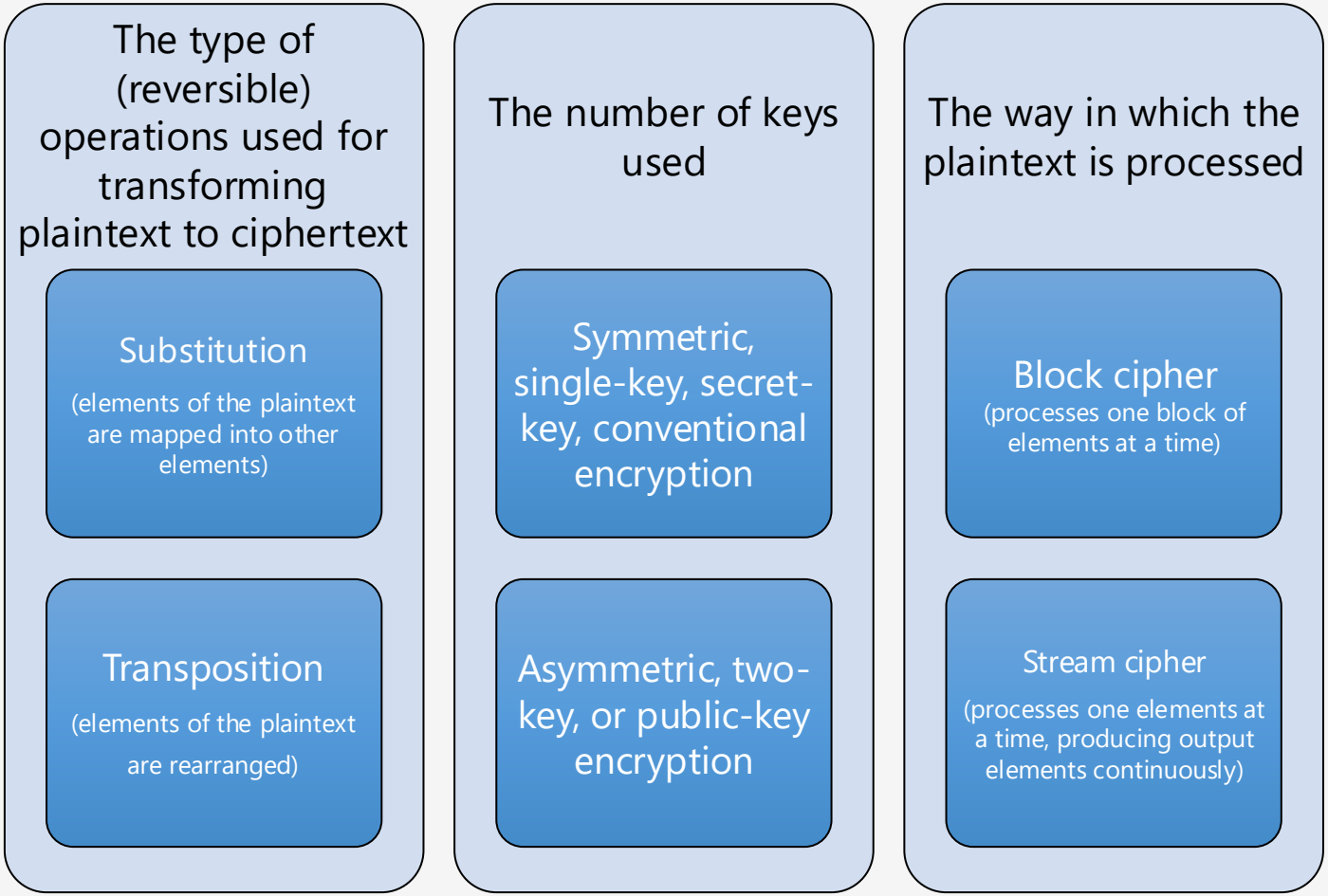


Figure 3.2 Model of Symmetric Cryptosystem

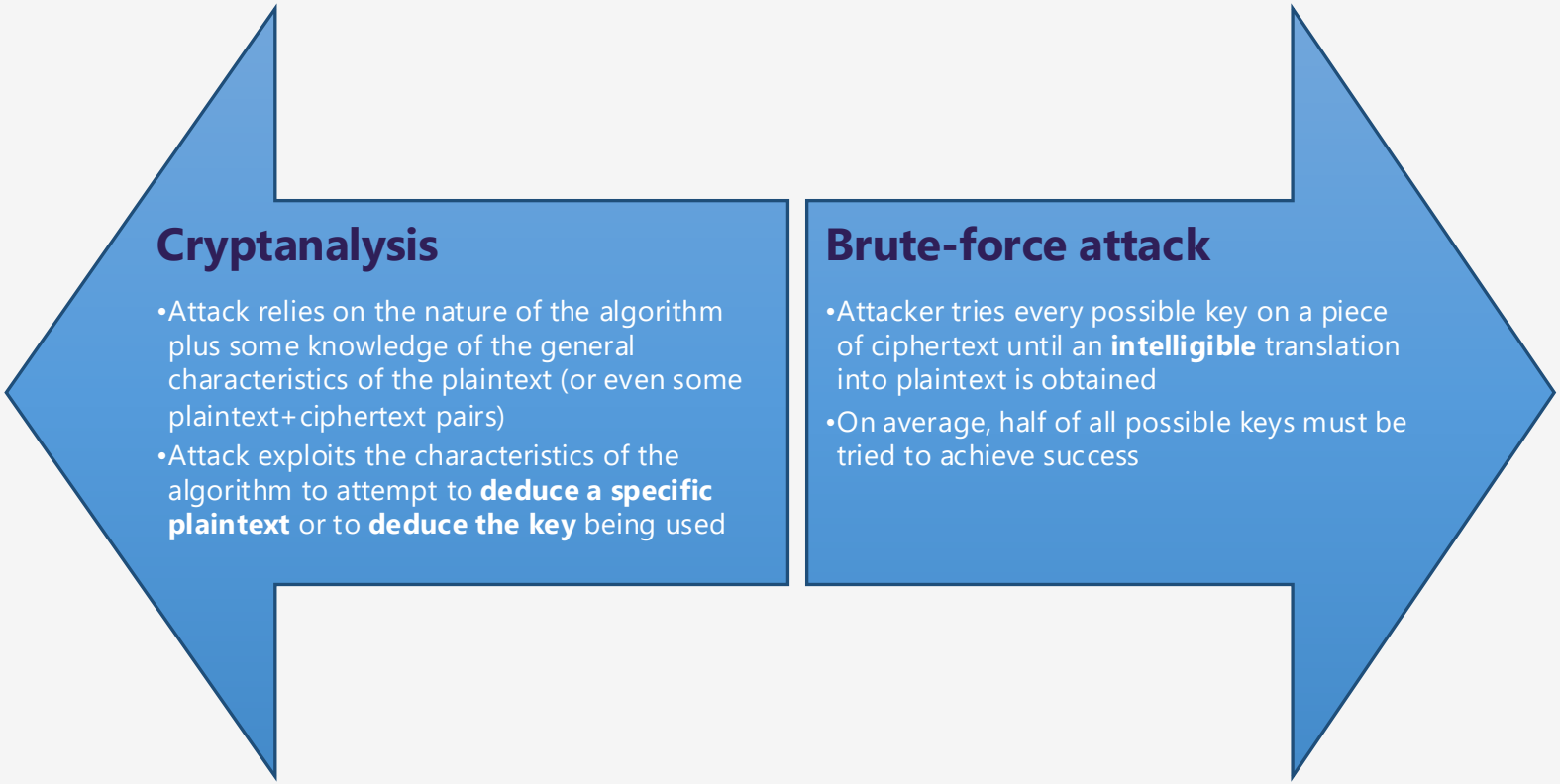
Cryptographic Systems

Cryptographic Systems are characterized along three independent dimensions:



Cryptanalysis and Brute-Force Attacks

- Two (very different) approaches to attacking a conventional encryption system
- If either of the approaches succeeds in finding the key, the effect is catastrophic: all future and past messages encrypted with that key are compromised



Brute-Force Attack

- Unless plaintext is provided, the analyst must be able to recognize plaintext as plaintext (e.g. if compressed before encryption this task is more difficult)

- Assuming a distributed setup or specialized hardware (like FPGAs or ASICs)

Cipher Name	Key Size (bits)	Number of Keys	Time to Brute-Force @ 10^{14} keys/sec
DES	56	7.2×10^{16}	~12 minutes
3DES (2-key)	112	5.2×10^{33}	~1.65 million years
3DES (3-key)	168	3.7×10^{50}	~1.17×10^{28} years
AES-128	128	3.4×10^{38}	~1.08×10^{24} years
AES-192	192	6.3×10^{57}	~2×10^{43} years
AES-256	256	1.2×10^{77}	~3.8×10^{63} years
Blowfish (128-bit)	128	3.4×10^{38}	~1.08×10^{24} years
RC4 (128-bit)	128	3.4×10^{38}	~1.08×10^{24} years

Type of Cryptanalytic Attacks (on Encrypted Messages)

- The most difficult problem is when only the ciphertext is available
- The brute-force approach is usually not feasible
- Attacker applies various statistical tests to the ciphertext, possibly with some general idea of the type of plaintext concealed
- An encryption algorithm is designed to withstand a **know-plaintext attack**, only weak algorithms fail to withstand a **ciphertext-only attack**

Type of Attack	Known to Cryptanalyst
Ciphertext Only	• Encryption algorithm • Ciphertext
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext-ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

T04 – Symmetric Encryption

Classical Encryption Techniques

Substitution Techniques

- The two basic building blocks of all encryption techniques are **substitution** and **transposition**, most systems combine the two

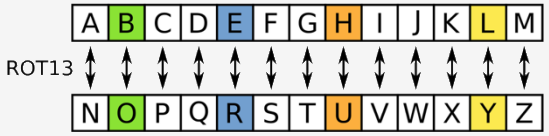
- The letters of plaintext are replaced by other letters or by numbers or symbols

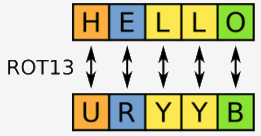
- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns

- The earliest known, and the simplest, use of a substitution cipher was by Julius Caesar

Substitution

(elements of the plaintext are mapped into other elements)





Transposition Techniques: Rail Fence

- Transposition Techniques perform some sort of permutation on the plaintext letters
- A simple example: Rail Fence Cipher
  - ✓ Plaintext is written down as a sequence of diagonals and then read off as a sequence of rows
  - ✓ To encipher the message "meet me tomorrow", we would write:

Transposition
(elements of the plaintext are rearranged)

m		e		m		t		m		r		o	
	e		t		e		o		o		r		w

Encrypted message is:

MEMTMROETEOORW

Caesar Cipher

- **Substitution** cipher used by Julius Caesar
- Involves replacing each letter of the alphabet with the letter standing three places further down the alphabet
- The alphabet is wrapped around (the letter following Z is A)
- Alphabet is wrapped around so that the letter following Z is A

plain:	meet	me	after	the	toga	party
cipher:	PHHW	PH	DIWHU	WKH	WRJD	SDUWB

- What is the plaintext of the following message?

cipher:	vwl	fodvvhv	duh	ixq
plain:	???	???????	???	???

https://cryptii.com/pipes/caesar-cipher

Breaking the Caesar Cipher is trivial..

- What is the "key" in this Cipher?
- Brute-force is very easy against this cipher:
  - The encryption/decryption algorithm are known
  - There are only 25 keys to try
  - The language of the plaintext is known and easily recognizable
  - On average we need 25/2 tries to obtain the original plaintext and find the key

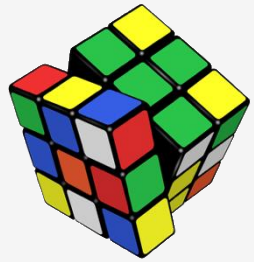
Key Size	Possible combinations
1-bit	2
2-bit	4
4-bit	16
8-bit	256
16-bit	65536
32-bit	4.2×10^9
56-bit (DES)	7.2×10^{16}
64-bit	1.8×10^{19}
128-bit (AES)	3.4×10^{38}
192-bit (AES)	6.2×10^{57}
256-bit (AES)	1.1×10^{77}

Note: What makes in general brute-force impractical is the usage of an algorithm that employs a large number of keys (**key size**!)

Monoalphabetic Ciphers

- A substitution cipher which uses a **fixed substitution** over the entire message; Special case: Caeser cipher (uses a fixed key)

- Permutation of a finite set of elements S:
  - An ordered sequence of all the elements of S, with each element appearing exactly once



- The term "monoalphabetic" indicates that there's only one substitution key used for the entire message.

- This type of cipher, while easy to understand and implement, is vulnerable to frequency analysis (with this technique ciphertext still reflects the frequency data of the original alphabet)

- If the "cipher" (encryption algorithm) can be any permutation of the 26 alphabetic characters, then there are 26! possible keys

  ✓ This is 10 orders of magnitude greater than the key space for DES
  ✓ But, much less secure!

Relative Frequency of Letters in English Text

- Cryptanalysis approach: compare the relative frequency of the letters in the ciphertext with the frequency of the letters in the English language
- A powerful tool is to look at the frequency of letter combinations (digrams, trigrams, …)
- Digram
  - ✓ Two-letter combination
  - ✓ Most common is *th* in the English language
- Trigram
  - ✓ Three-letter combination
  - ✓ Most frequent is *the* in the English language

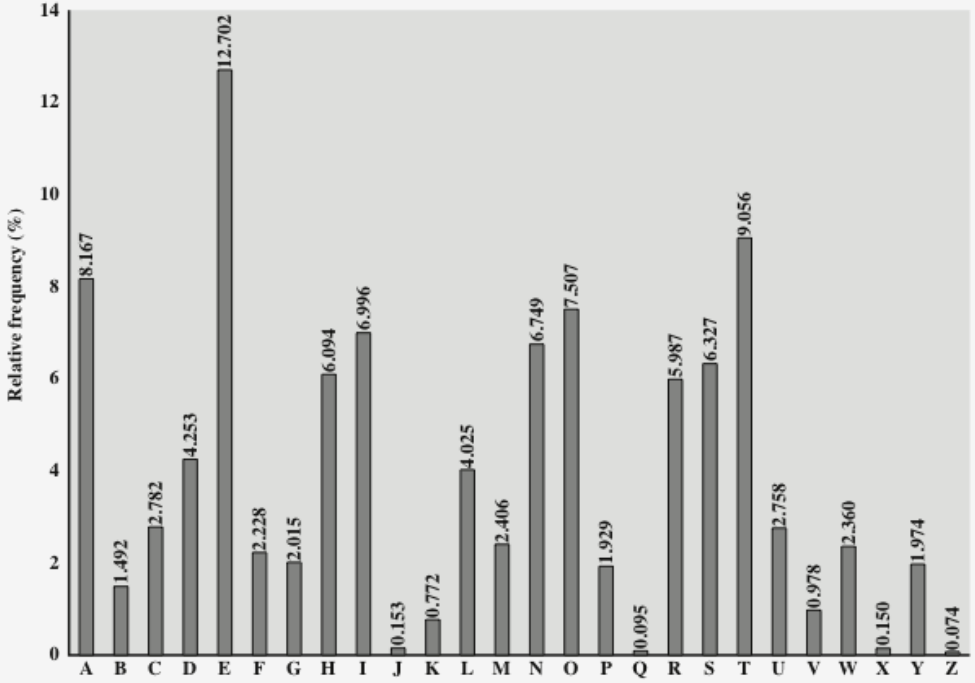


Figure 3.5 Relative Frequency of Letters in English Text

Polyalphabetic Ciphers

- Polyalphabetic substitution cipher, the best known is the Viginère Cipher
- Improves on the simple monoalphabetic technique by using different monoalphabetic substitutions as one proceeds through the plaintext message

All these techniques have the following features in common:

- A set of related monoalphabetic substitution rules is used
- A key determines *which particular rule is chosen for a given transformation*

Vigenère Cipher

- Best known and one of the simplest polyalphabetic substitution ciphers
- In this scheme the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers with shifts of 0 through 25
- Each cipher is denoted by a key letter which is the ciphertext letter that substitutes for the plaintext letter ‘a’
- To encrypt a message, a key is needed that is as long as the message
- Usually, the key is a repeating keyword

Vigenère Autokey System

- A keyword is concatenated with the plaintext itself to provide a running key

  Example:

key:	deceptivewearediscoveredsav
plaintext:	wearediscoveredsaveyourself
ciphertext:	ZICVTWQNGKZEIIGASXSTSLVVWLA

- The Vigenère cipher is more resistant to frequency analysis compared to simple monoalphabetic substitution ciphers because it uses a keyword to determine the shift applied to each letter in the plaintext. However, it is not entirely immune to frequency attacks.

- Because the key and the plaintext share the same frequency distribution of letters, a statistical technique can be applied

- To enhance the security of the Vigenère cipher, techniques such as using a longer, truly random keyword or employing additional steps like the use of a one-time pad can be utilized.

One-Time Pad

- Use a **random** key that is as long as the message so that the key need not be repeated
- Key is used to encrypt and decrypt a single message and then is discarded
- Each new message requires a new key of the same length as the message
- Exhibits **perfect secrecy:**
  - Given an encrypted message (or ciphertext) from a perfectly secure encryption system (or cipher), absolutely nothing will be revealed about the unencrypted message (or plaintext) by the ciphertext
  - The ciphertext conveys no information about the content of the plaintext.
  - It can be proved that any such scheme must use at least as much key material as there is plaintext to encrypt.
  - The probability distribution of the possible plaintexts is independent of the ciphertext.

One-Time Pad: difficulties

- The One-Time Pad scheme is **unbreakable** and offers complete security, but:
  - ✓ There is the practical problem of making large quantities of random keys
    - ➢ Any heavily used system might require millions of random characters on a regular basis
    - ➢ Truly versus "pseudo" random number generators
  - ✓ Enormous key distribution problem
    - ➢ For every message to be sent, a key of equal length is needed by both sender and receiver
- Because of these difficulties, the one-time pad is of limited utility
  - ✓ Useful primarily for low-bandwidth channels requiring very high security (e.g diplomatic channels, offline digital security,..)

Rotor Machines

- A polyalphabetic substitution algorithm with a period of 26 for each cylinder
- With three cylinders: 26x26x26 = 17576 different substitution alphabets
- Rotor machines point the way to a large class of symmetric ciphers, of which DES is a representative

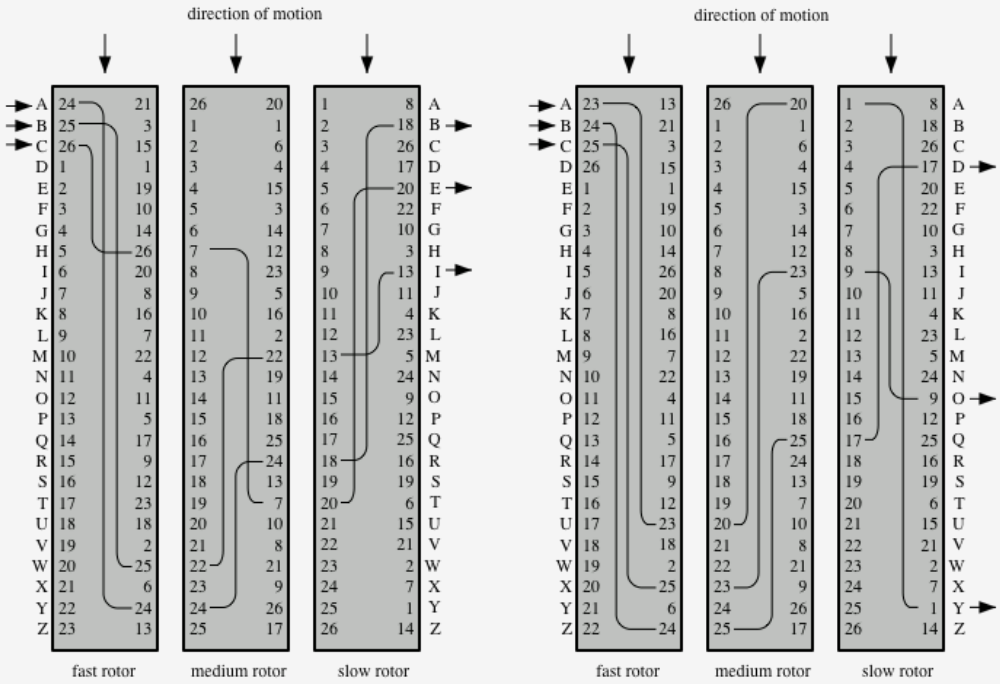


Figure 3.8 Three-Rotor Machine With Wiring Represented by Numbered Contacts

The enigma machine

https://www.youtube.com/watch?v=-qcOCBfRRzg

Steganography

- **Conceal** the existence of the message, rather than render the message unintelligible
- It can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered
- A classic example:
  - Invisible ink: a number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper

Steganography (hidden text in images)

- For example, in an image with resolution 3094 x 6144 using 24 bits per pixel, the least significant pixel can be change without greatly affecting the quality of the image
- In this example, a 130 kB message can be hidden in the image

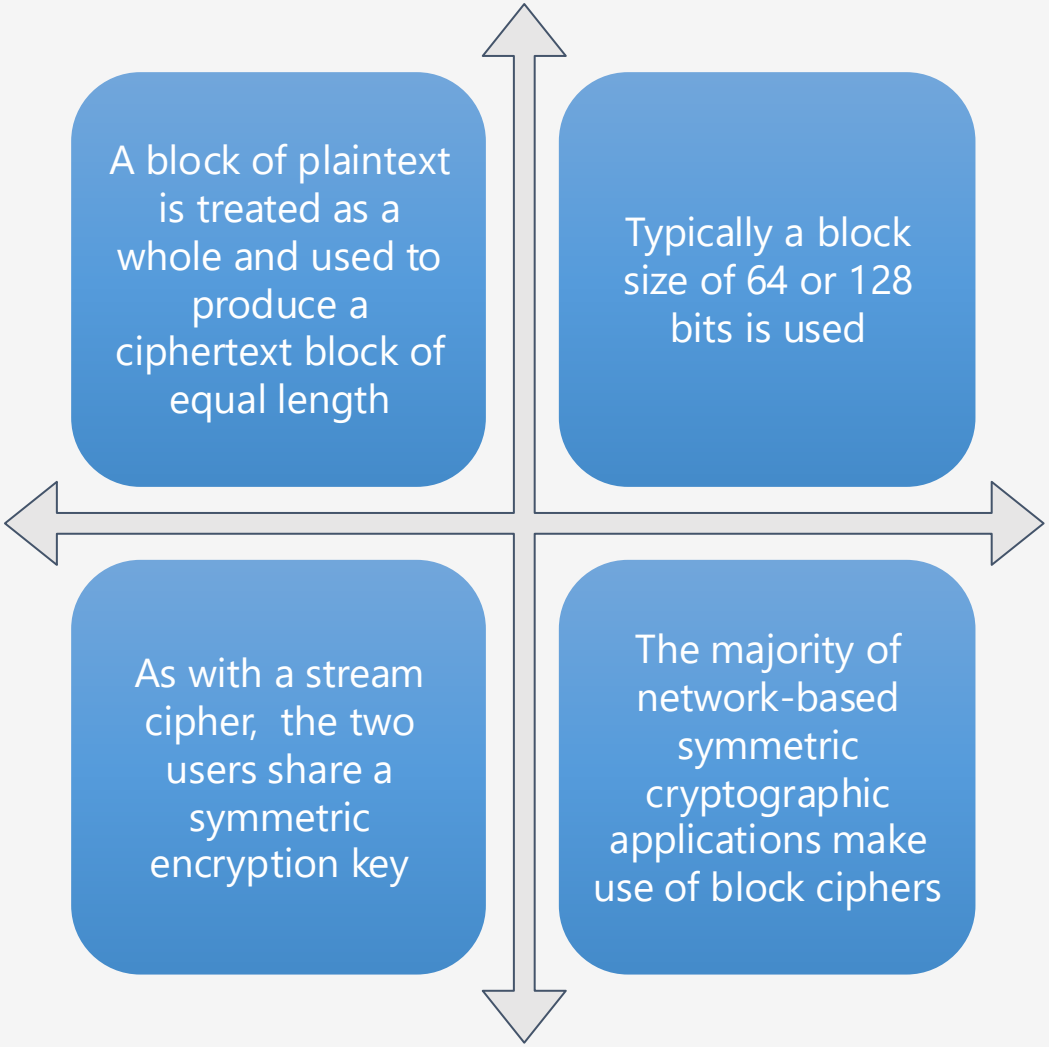
https://incoherency.co.uk/image-steganography/

T04 – Symmetric Encryption

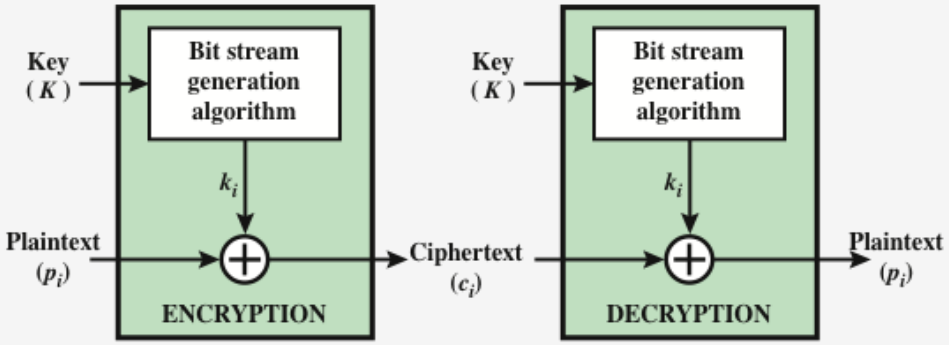
Block ciphers + Feistel design

Modes of operation

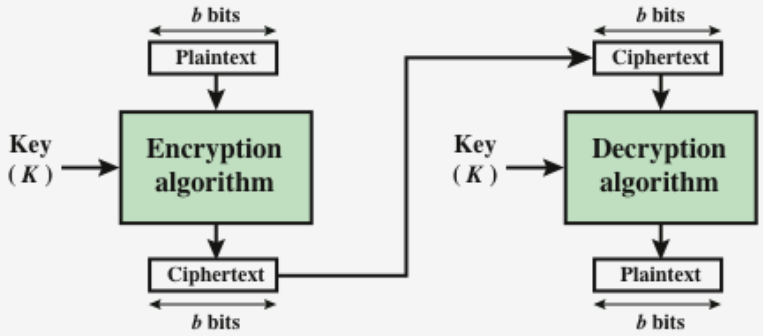
Block Ciphers



Stream Ciphers vs. Block Ciphers



(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

Figure 4.1 Stream Cipher and Block Cipher

Feistel Cipher

- Most symmetric block ciphers are based on a **Feistel Cipher Structure**
- Feistel proposed the use of a cipher that alternates substitutions and permutations
- Is the structure used by many significant symmetric block ciphers currently in use

Substitutions

- Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements

Permutation

- No elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed

Diffusion and Confusion

- Terms introduced by Claude Shannon to capture the two basic building blocks for any cryptographic system
- Shannon's concern was to thwart cryptanalysis based on statistical analysis
- These principles have become the cornerstone of modern block cipher design

Diffusion

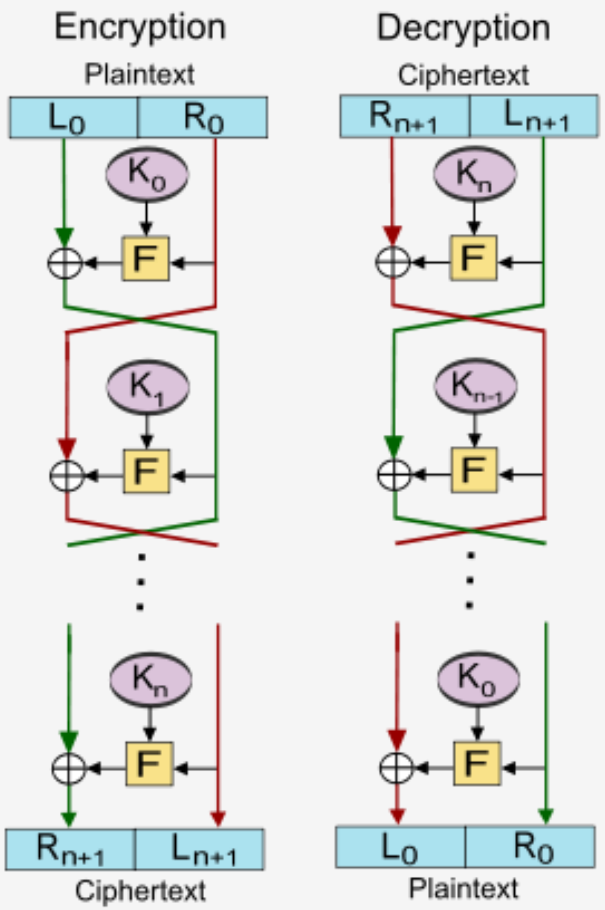
- The statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext
- This is achieved by having each plaintext digit affect the value of many ciphertext digits

Confusion

- Seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible
- Even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key

Feistel network

- A Feistel network is a cryptographic structure used in the construction of block ciphers.
- In a Feistel network, the input block is divided into two equal parts, which are then subjected to a series of rounds of substitutions and permutations, using a set of keys. At the end of the rounds, the two halves are combined to produce the output block.
- The Feistel design allows for the creation of complex encryption functions using simple and easily reversible operations, such as substitutions and permutations.
- Many well-known block ciphers, such as DES, AES, and Blowfish, are based on this design.



Feistel Encryption and Decryption

- A substitution is performed on the left half of the data (using round function F)
- F depends on the algorithm, usually involves expansion, combination with subkey, substitution (S-boxes in DES) for confusion, and permutation
- A permutation is performed that consists of the interchange of the two halves of the data (diffusion)
- Same process supports encryption and decryption (facilitates implementations in software and hardware)

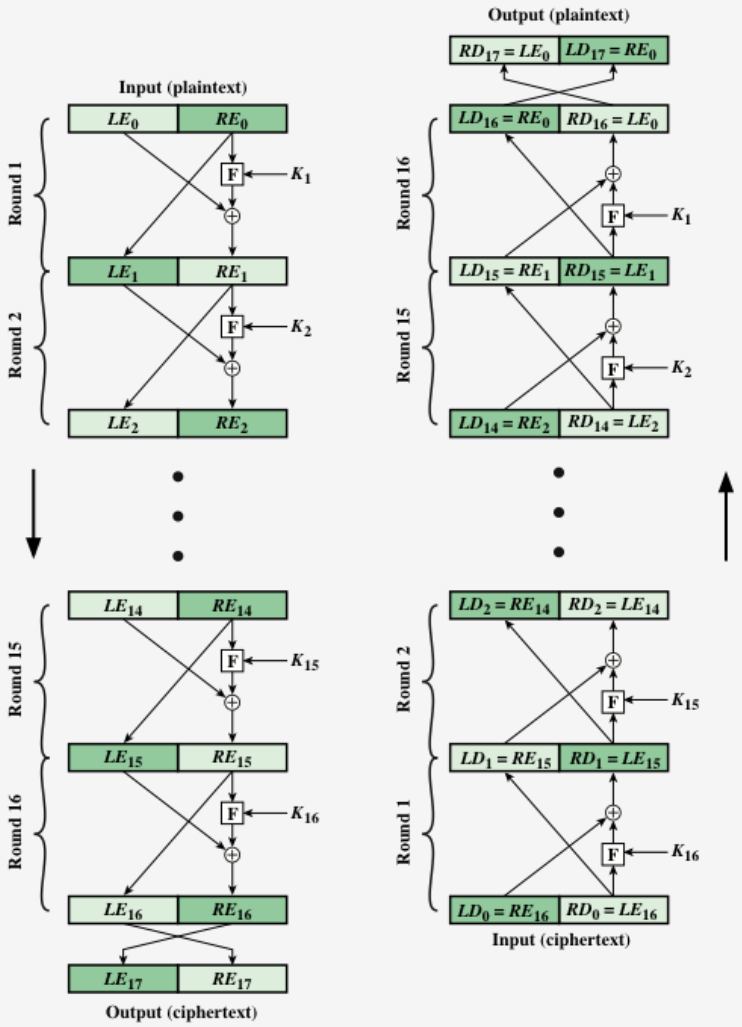


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

Feistel Cipher Design Features

Block size

- Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm

Key size

- Larger key size means greater security but may decrease encryption/decryption speeds

Number of rounds

- The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security

Subkey generation algorithm

- Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis

Ease of analysis

- If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength

Modes of operation (of symmetric ciphers)

- Symmetric ciphers are used in different modes to provide a variety of encryption services that meet different security requirements and operational needs.
- One reason for using different modes is to provide confidentiality for data of different lengths.
- To apply a block cipher in a variety of applications, five modes of operation have been defined by NIST (SP 800-38A): ECB, CBC, CTR, CFB, OFB
- These modes are intended for use with any **symmetric** block cipher, including 3DES and AES

Block Cipher Modes of operation (summary)

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of plaintext bits is encoded independently using the same key.	•Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next block of plaintext and the preceding block of ciphertext.	•General-purpose block-oriented transmission •Authentication
Cipher Feedback (CFB)	Input is processed *s* bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	•General-purpose stream-oriented transmission •Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	•Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	•General-purpose block-oriented transmission •Useful for high-speed requirements

Electronic Codebook (ECB)

Each block of plaintext bits is encoded independently using the same key.	•Secure transmission of single values (e.g., an encryption key)

- Each block of plaintext is independently encrypted using the same key
- Simple and easy to implement, easily parallelized for high-speed encryption
- Useful only to secure messages shorter than a single block of ciphertext (e.g. a secret key): 64 bits for 3DES, 128 bits for AES
- Identical plaintext blocks produce identical ciphertext blocks, which can reveal patterns in the plaintext and make the encryption vulnerable to certain attacks (lacks diffusion)
- The use of a single key for all blocks makes ECB mode vulnerable to key replay attacks, where an attacker can simply replay a block of ciphertext with the same key to obtain the corresponding plaintext.

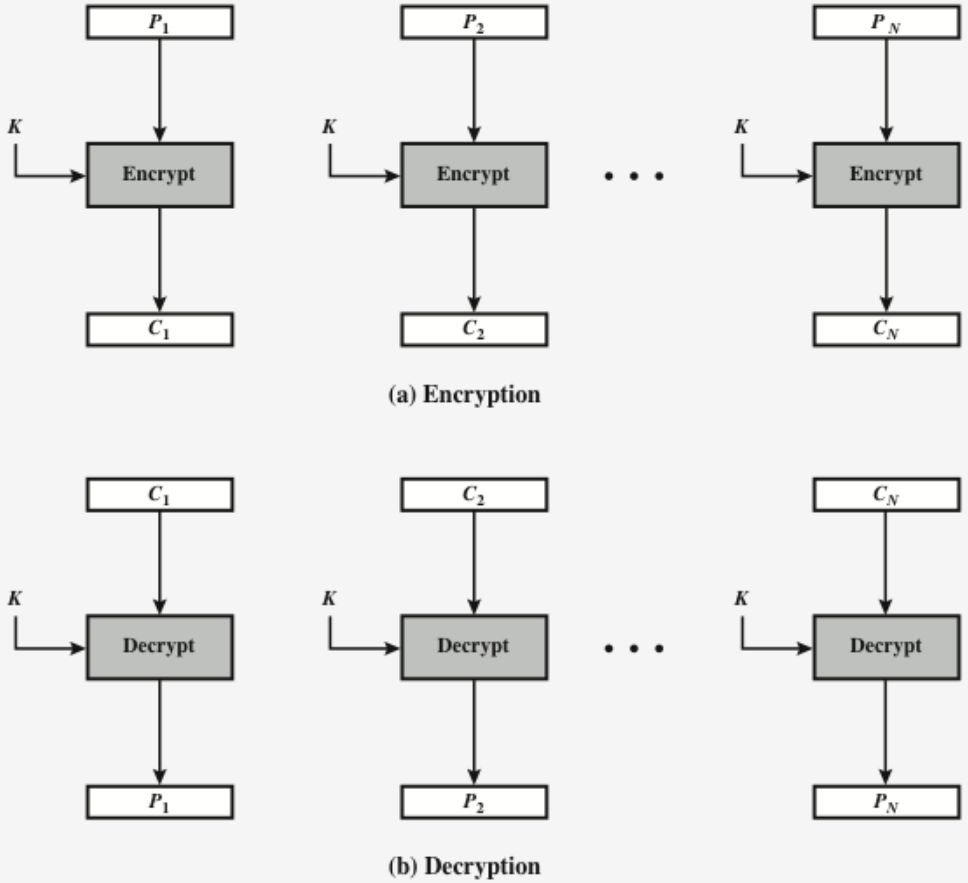
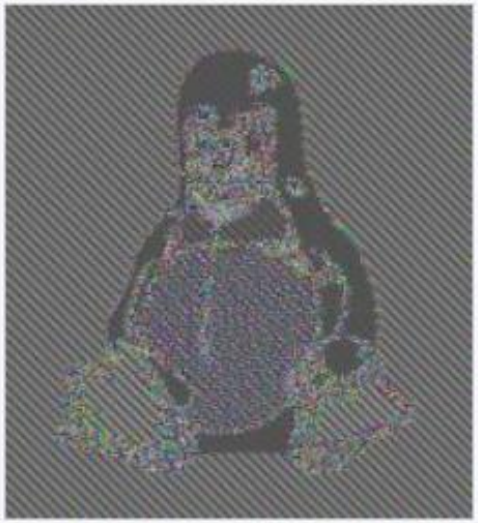


Figure 7.3 Electronic Codebook (ECB) Mode

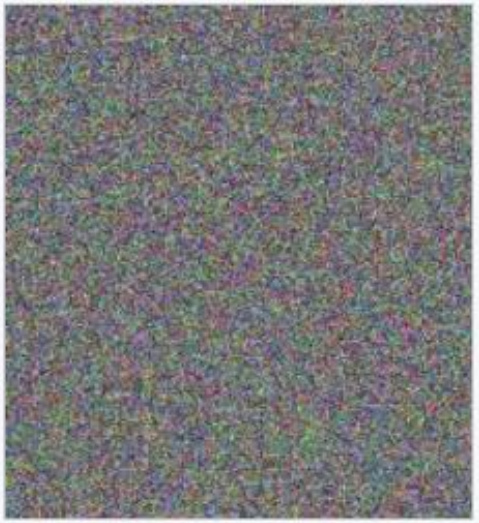
Electronic Codebook (ECB)

- ECB lacks diffusion, because it encrypts identical plaintext blocks into identical ciphertext blocks
- It does not hide data patterns well
- Example for a bitmap image: the color of each individual pixel is encrypted, but the overall image may still be discerned

Original image



Encrypted using ECB mode



Modes other than ECB result in pseudo-randomness

Being ECB our "baseline", how may we evaluate superior modes?

- Overhead: additional operations for encryption and decryption
- Error recovery: ability of a block cipher mode of operation to detect and recover from errors that may occur during the transmission or storage of encrypted data
- Error propagation: a characteristic of some block cipher modes of operation that describes how errors in the ciphertext propagate to the decryption of subsequent blocks
- Diffusion: how the plaintext statistics are reflected in the ciphertext

Cipher Block Chaining (CBC)

The input to the encryption algorithm is the XOR of the next block of plaintext and the preceding block of ciphertext.	•General-purpose block-oriented transmission •Authentication

- The same plaintext block, if repeated, produces different ciphertext blocks
- The same key is used for each block
- Encryption is sequential, cannot be parallelized
- Decryption can be parallelized (a plaintext block is obtained from two adjacent ciphertext blocks)
- A one-bit change in a plaintext or IV affects all following ciphertext blocks
- Decrypting with the incorrect IV only affects the first plaintext block
- Used in situations where confidentiality and integrity of data are both important.

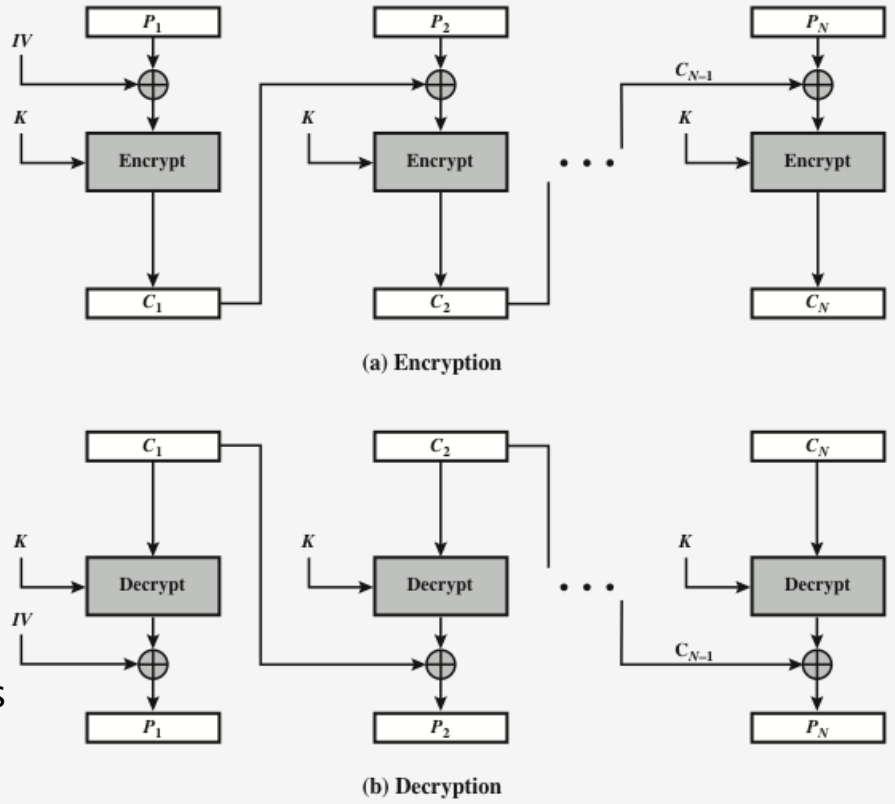


Figure 7.4 Cipher Block Chaining (CBC) Mode

Cipher Feedback (CFB)

Input is processed s bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.

- General-purpose stream-oriented transmission
- Authentication

- As with CBC, the input block to each forward cipher function depends on the result of the previous function
- Oriented towards stream encryption, because it operates on individual bytes or bits of the plaintext, rather than on fixed-size blocks.
- Encryption cannon be parallelized
- Like CBC, changes in plaintext propagate forever in the ciphertext, and decryption can be parallelized
- Advantage over CBC: block cipher only used in encryption mode, also does not need padding in last block

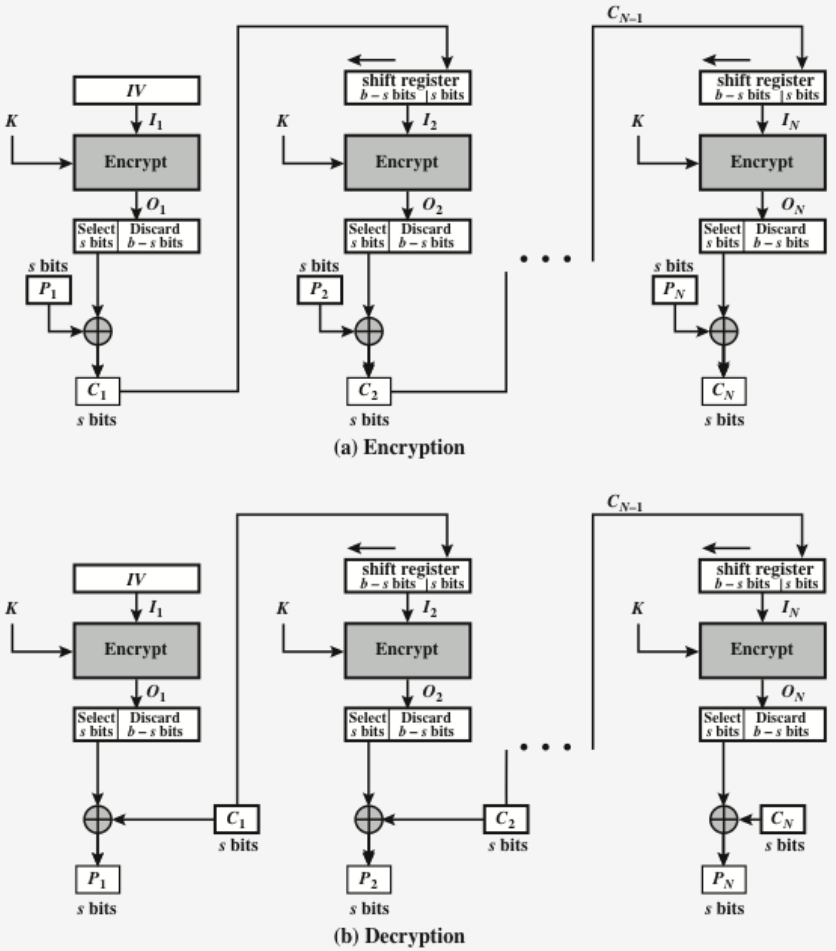


Figure 7.5 *s*-bit Cipher Feedback (CFB) Mode

Output Feedback (OFB)

Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.

•Stream-oriented transmission over noisy channel (e.g., satellite communication)

- Makes a block cipher into a stream cipher
- Generates keystream blocks, which are then XORed with the plaintext
- Operates on full blocks of plaintext and ciphertext
- As with CBC and CFB, also requires an IV unique to each encryption (a cryptographic nonce: an arbitrary number used only once)
- Bit errors in transmission do not propagate (only affect the corresponding recovered plaintext block)

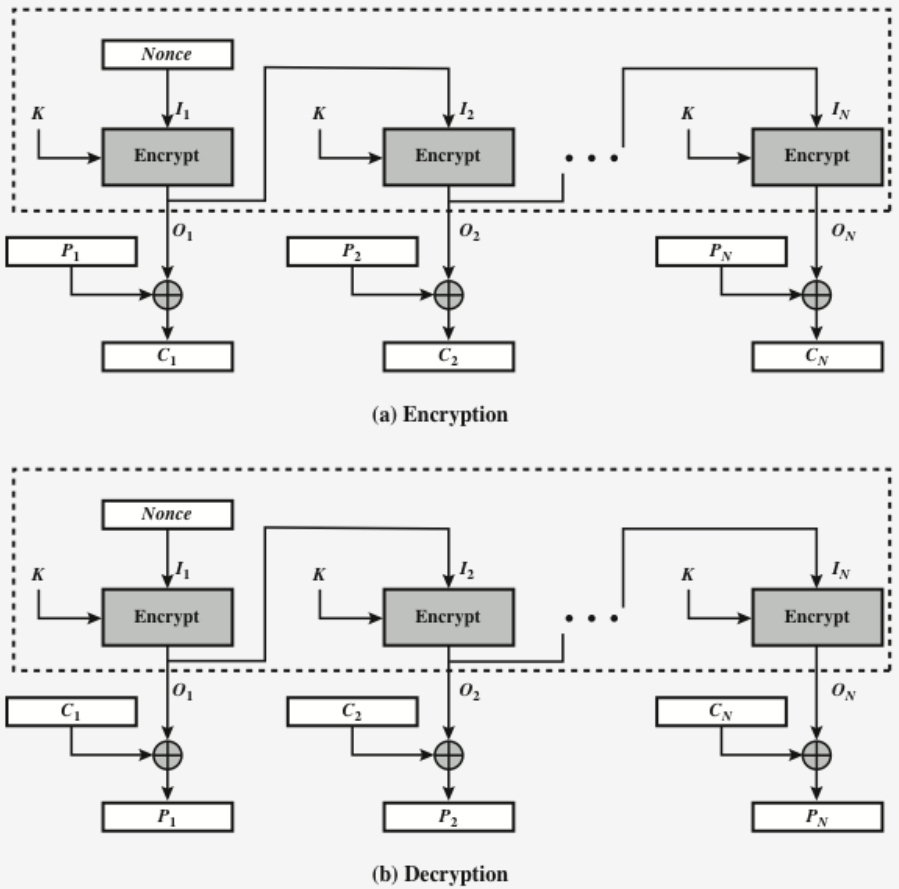


Figure 7.6 Output Feedback (OFB) Mode

Counter (CTR) Mode

Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.

- General-purpose block-oriented transmission
- Useful for high-speed requirements

- Generates next keystream block by encrypting successive values of a "counter"
- Counter needs to be different for each encrypted plaintext block, typically it is initialized and then incremented for each block
- As with OFB, the initial counter value must be a *nonce*
- Efficient in hardware and software: CTR can perform in parallel on multiple blocks of plaintext/ciphertext
- Simplicity in implementation, as it requires only an encryption algorithm

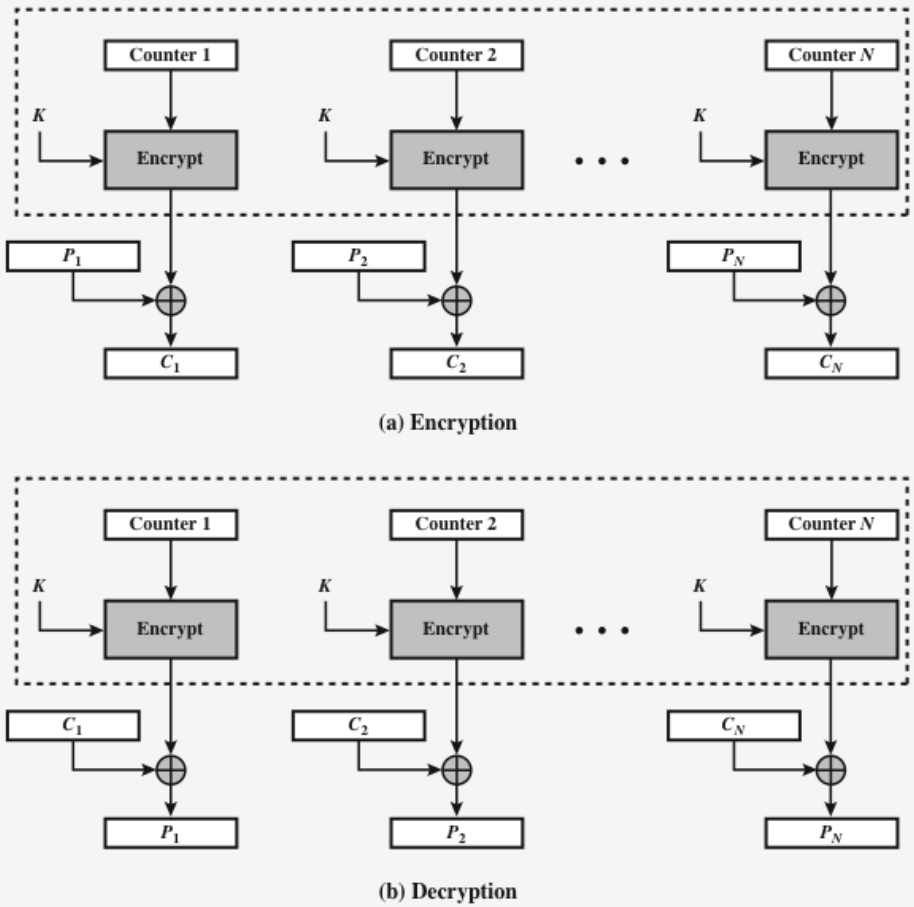


Figure 7.7 Counter (CTR) Mode

Modes of operation (IV - Initialization Vector)

- IV (Initialization Vector): a block of bits used by several modes to randomize the encryption and hence produce distinct ciphertexts, even if the same plaintext is encrypted multiple times. Required with CBC, CFB, OFB and CTR
- Some modes (ECB and CBC) also require that the last block be padded before encryption, if required by the application
- While the IV doesn't necessarily need to be kept secret, it should be chosen randomly and kept unique for each message encrypted with the same key to ensure the security of the encryption.
- The use of a random and unique IV is important for ensuring the security of block cipher modes, as it helps to prevent certain types of attacks that rely on repeating patterns in the ciphertext.

Which mode to use?

It's important to use a mode that provides good security and meets the specific needs of the application. Some block cipher modes have specific advantages or disadvantages depending on the application, and may be more or less appropriate for certain use cases. For example:

- ECB (Electronic Codebook) mode is generally considered to be the least secure block cipher mode, as it should only be used when encrypting very small amounts of data, or when compatibility with legacy systems requires its use.
- CBC (Cipher Block Chaining) mode is a widely used and well-regarded block cipher mode, as it provides good diffusion and error propagation, and is resistant to most types of attacks.
- CTR (Counter) mode is also a popular choice, as it provides good security and can be easily parallelized for efficient implementation (e.g. IoT devices). It doesn't require padding, and can be used to encrypt and decrypt data in a streaming fashion.
- OFB and CFB can also be a good choice for applications that require streaming encryption and don't require padding.

Which mode to use?

Encryption Mode	Advantages	Disadvantages	Example Applications
ECB (Electronic Codebook)	- Simple to implement - Allows parallel encryption of blocks	- Does not hide patterns in data - Insecure for structured or repetitive data	- Encrypting small, patternless data - Temporary or random data
CBC (Cipher Block Chaining)	- Hides patterns in data - Good block integrity	- No parallelism in encryption - Error propagation (one error affects subsequent blocks)	- File or backup encryption - TLS (in older versions)
CTR (Counter Mode)	- Fully parallelizable - Converts block cipher into stream cipher - No error propagation	- Reusing the same counter (nonce) breaks security - Requires careful synchronization	- IoT and embedded systems - VPNs - Secure cloud storage
OFB (Output Feedback)	- Converts block cipher into stream cipher - No transmission error propagation	- Vulnerable to IV reuse - No data authentication (no integrity checking) - Less efficient than CTR	- Legacy systems - Niche cases with compatibility constraints

T04 – Symmetric Encryption

Symmetric block encryption algorithms:

DES, 3DES, AES

Data Encryption Standard (DES)

- Issued in 1977 by the National Bureau of Standards (now NIST) as Federal Information Processing Standard 46
- Was the most widely used encryption scheme until the introduction of the Advanced Encryption Standard (AES) in 2001
- 3DES was recommended instead in 1999 (3DES applies DES 3 times to the same plaintext, using different keys to produce ciphertext)
- Algorithm itself is referred to as the Data Encryption Algorithm (DEA)
  - ✓ Data are encrypted in 64-bit blocks using a 56-bit key
  - ✓ The algorithm transforms 64-bit input in a series of steps into a 64-bit output
  - ✓ The same steps, with the same key, are used to reverse the encryption

DES Encryption Algorithm

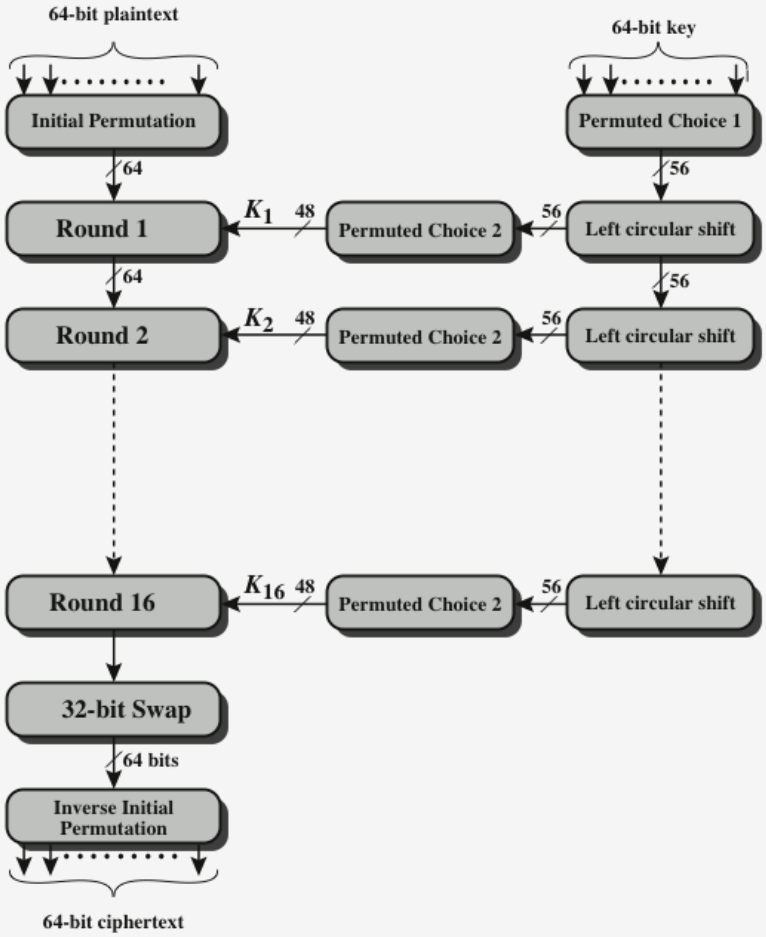


Figure 4.5 General Depiction of DES Encryption Algorithm

Round	*Ki*	*Li*	*Ri*
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP–1		da02ce3a	89ecac3b

- Used an Initial permutation (IP) and final inverse IP
- 16 rounds of the same function, which involves both permutation and substitutions
- PC-1 and PC-2 use fixed permutation tables defined in the algorithm
- The key is passed through an initial permutation function, for each round a different subkey is produced
- With the exception of the IP and inverse IP, DES has the exact structure of a Feistel cipher

Feistel structure and DES

- The 16 rounds of DES follow the Feistel structure
- In each Round, the plaintext is divided into two 32-bit blocks, known as the left half and right half.
- Each round applies the same function to the right half, using a subkey that is derived from the original encryption key. The result of this function is then XORed with the left half, and the two halves are swapped before the next round begins.
- After the 16 rounds have been completed, the final 64-bit output is subjected to a final permutation, known as the inverse initial permutation (IP^-1).

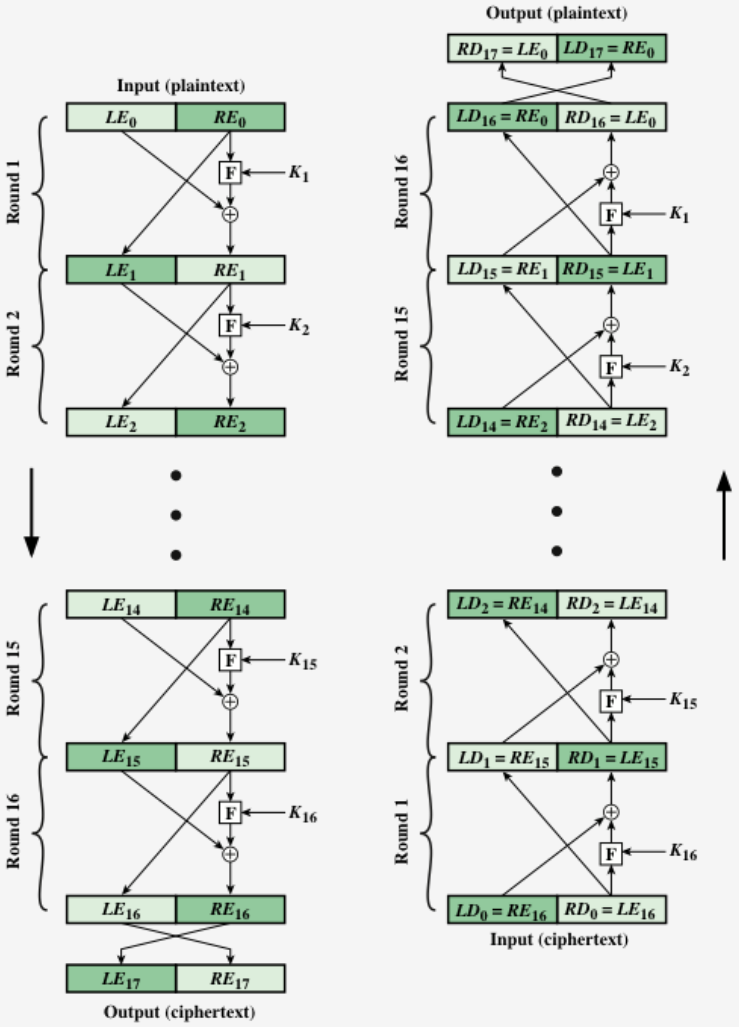
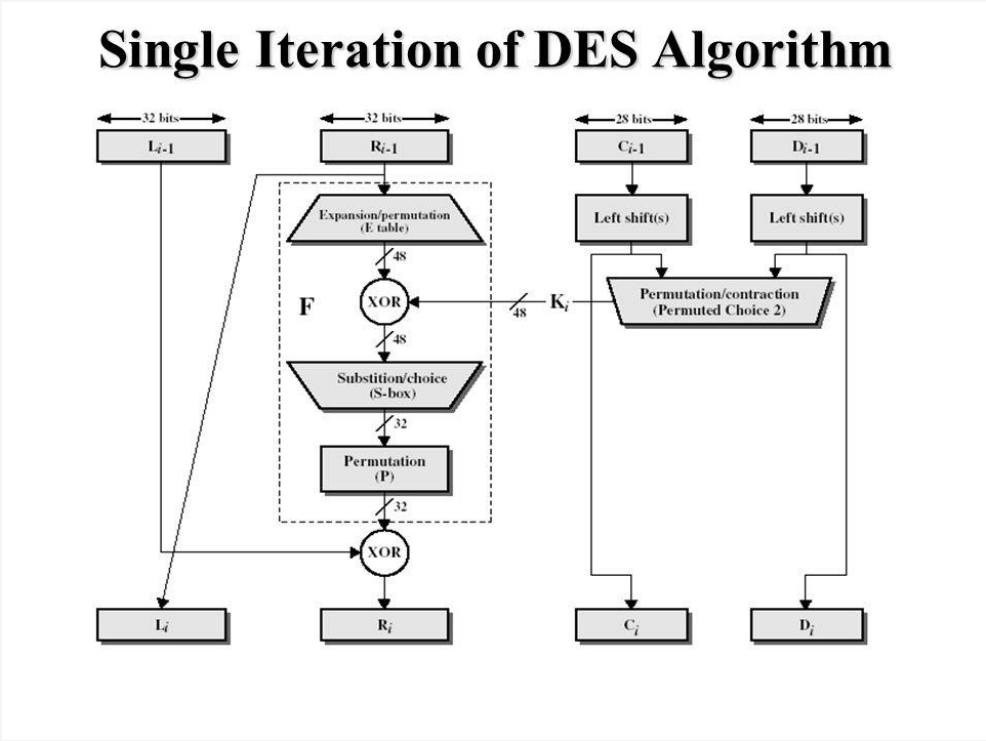


Figure 4.3 Feistel Encryption and Decryption (16 rounds)

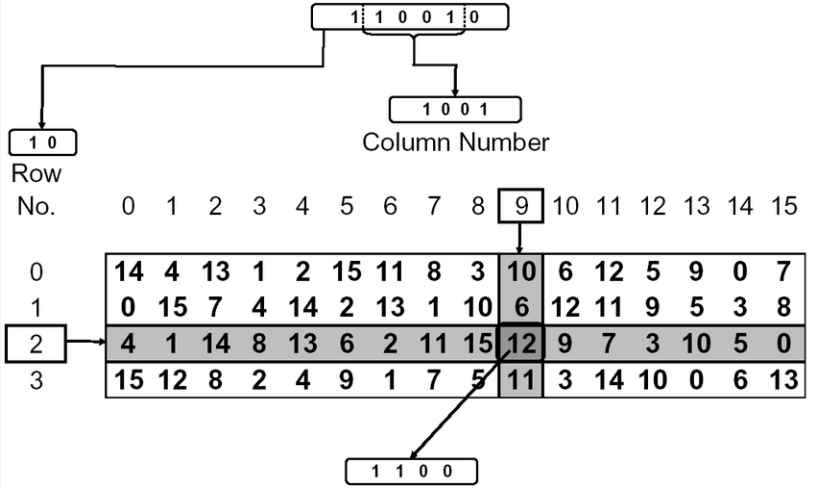
DES (a Round in more detail)

- Expansion/permutation: used in the Feistel network to expand a 32-bit input block into a 48-bit block for use in the encryption process.
- The E-Table consists of 48 entries that specify the positions of each bit in the expanded output block. The table is defined such that each bit in the 32-bit input block is mapped to several bits in the 48-bit output block.
- Substitution (S-boxes): Each round uses eight different substitution boxes (S-boxes) to further obscure the plaintext. The right half of the 64-bit block is divided into eight 6-bit sections, which are then used to look up values in the S-boxes. The resulting 32-bit output is used as input to the next round.
- Permutation (P-box): Finally, each round applies a permutation algorithm to the 32-bit output from the S-boxes. This is known as the P-box permutation, and it is different for each round.



DES (substitution and permutation tables)

- There are 8 S-boxes used in the DES algorithm, each containing 4 rows and 16 columns, and each taking a 6-bit input as an index to retrieve a 4-bit output. The S-boxes are applied to the output of the expansion operation in the Feistel network.
- The substitution operation performed by the S-boxes is a nonlinear operation that adds to the security of the DES algorithm. Each S-box is designed to have no mathematical structure, making it difficult to analyze and break the encryption.
- A permutation table is a predefined table used to perform a permutation operation on the input data.
- There are several permutation tables used in DES, including the Initial Permutation (IP) table, the Expansion (E) table, and the Permutation (P) table.



The Initial Permutation: IP

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

Avalanche Effect in DES (1-bit Change in Plaintext)

- A desirable property of any encryption algorithm is that a small change (e.g. one bit) in either the plaintext or the key should produce a significant (many bits) change in the ciphertext
- If the change were small, this might provide a way to reduce the size of the plaintext or key space to be searched

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP–1	da02ce3a89ecac3b 057cde97d7683f2a	32

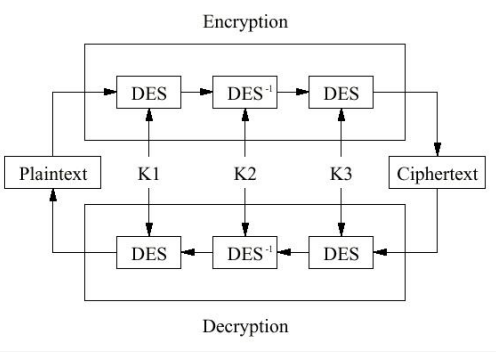
Avalanche Effect in DES (1-bit Change in Key)

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

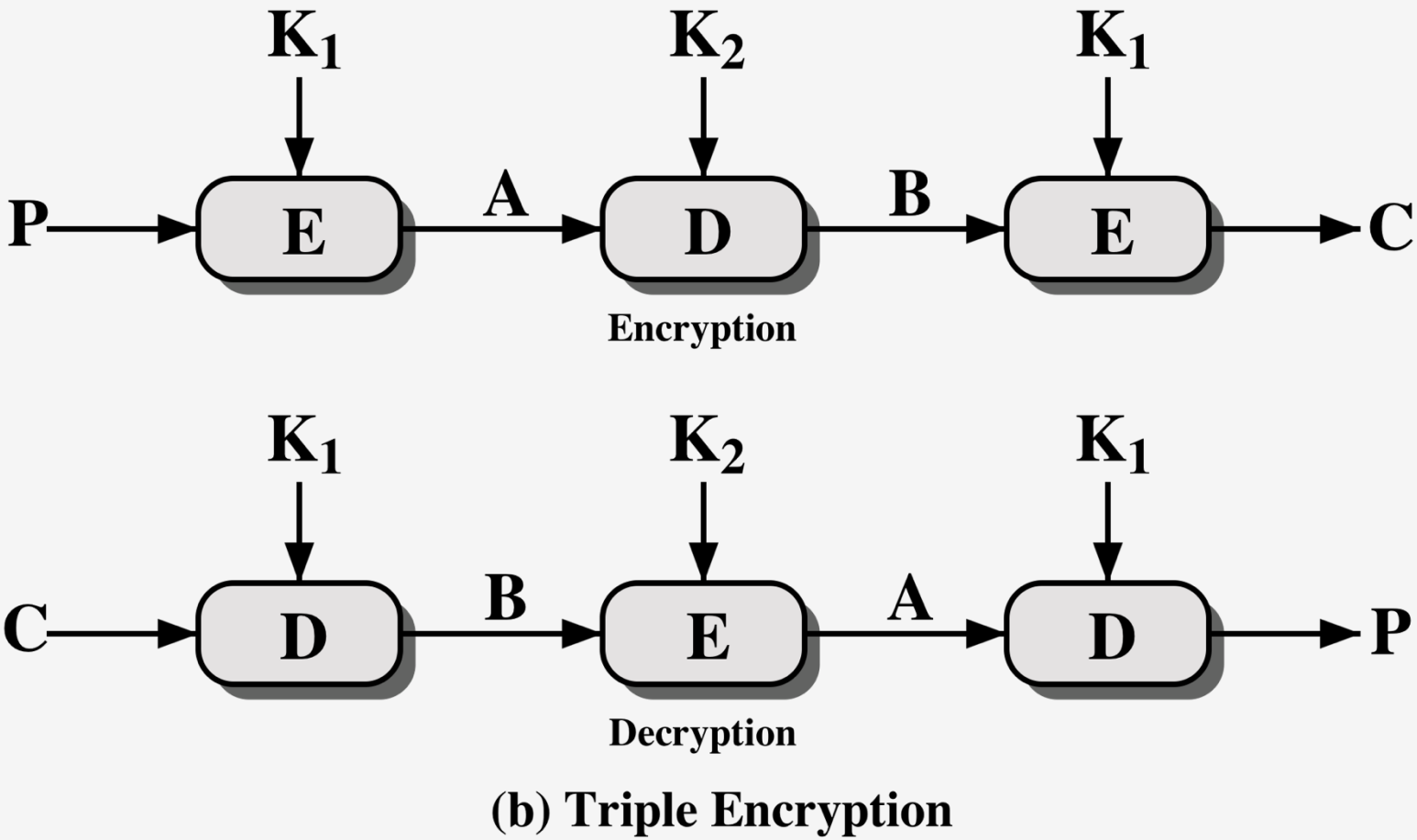
Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30

Multiple encryption and Triple DES (3DES)

- DES is vulnerable to brute-force attacks, and it has been largely replaced by stronger encryption schemes
- Two approaches have been taken:
  - Design a completely new algorithm (resistant to brute-force and cryptanalytic attacks): **AES**
  - Preserve existing investments in software and equipment using multiple encryption with DES and multiple keys: triple DES (**3DES**)



Triple encryption



(b) Triple Encryption

Triple DES using three Keys

Three-key DES (3DES) is a viable alternative to DES

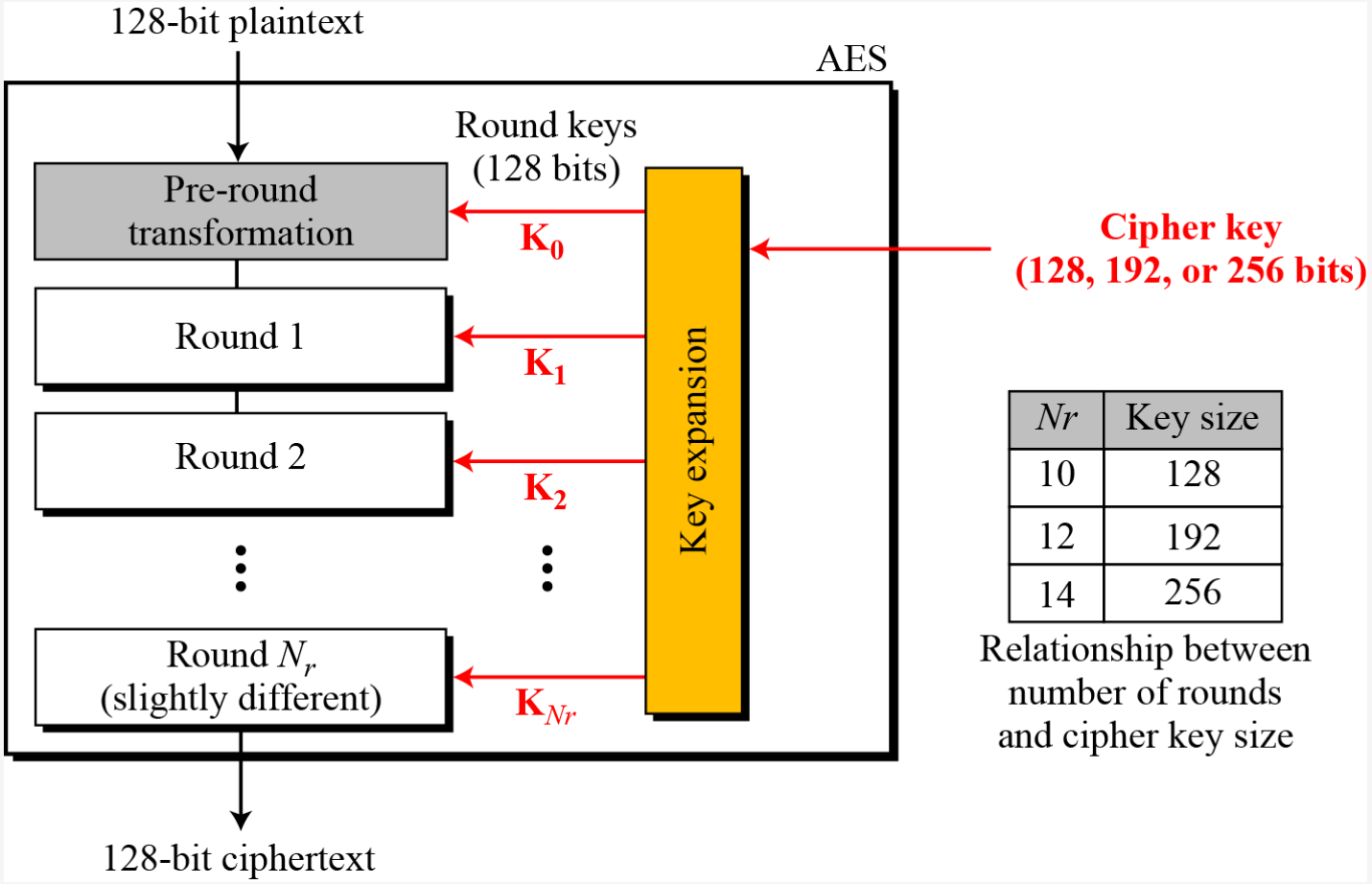
Three-key 3DES has an effective key length of 168 bits and is defined as:	• $C = E(K_3, D(K_2, E(K_1, P)))$
Backward compatibility with DES is provided by putting:	• $K_3 = K_2$ or $K_1 = K_2$

A number of Internet-based applications have adopted three-key 3DES, including PGP and S/MIME

Advanced Encryption Standard (AES)

- The Advanced Encryption Standard (AES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST) in December 2001, as a DES successor
- The criteria defined by NIST for selecting AES fall into three areas:
  - ✓ 1: Security - 128 bit key
  - ✓ 2: Cost – computational efficiency, storage requirement
  - ✓ 3: Implementation - flexibility and simplicity (implementable on any platform)
- AES encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, depends on the number of rounds.

General design of AES



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Security of AES

- AES was designed after DES. Most of the known attacks on DES were already tested on AES.
- Brute-Force Attack: AES is definitely more secure than DES due to the larger-size key.
- Statistical Attacks: Numerous tests have failed (so far) to do statistical analysis of the ciphertext.
- Differential cryptanalysis attacks (by exploring how differences in information input can affect the resultant difference at the output): there are no differential attacks on AES (as yet).
- Linear cryptanalysis attacks (recover the secret key used in a cipher by analyzing patterns in plaintext and ciphertext pairs.): there are no linear attacks on AES (as yet).

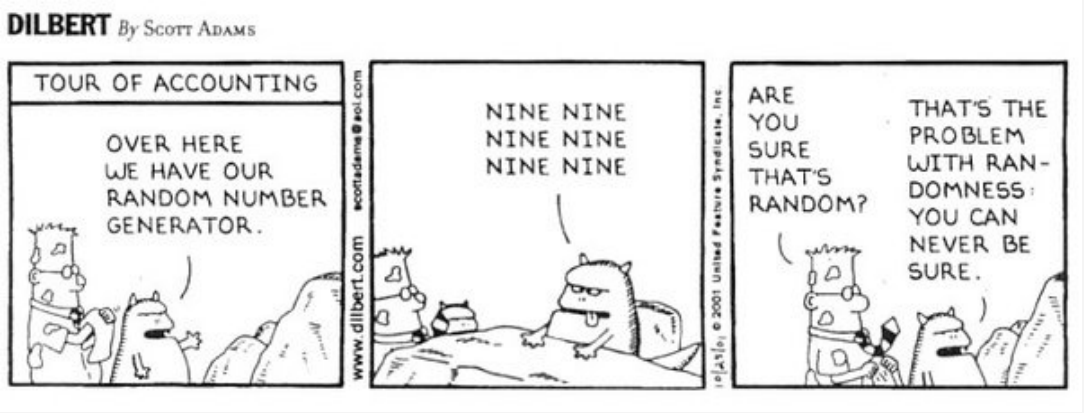
T04 – Symmetric Encryption

Random Bit Generation

Stream Ciphers

Pseudorandom Numbers

- Cryptographic applications typically make use of algorithmic techniques for random number generation
- These algorithms are deterministic and therefore produce sequences of numbers that are not statistically random
- If the algorithm is good, the resulting sequences will pass many tests of randomness and are referred to as pseudorandom numbers
- The use of pseurorandom numbers is widely accepted in cryptography



True vs. Pseudorandom Number generators

- TRNG takes as input a source that is effectively random (entropy source), e.g: mouse or keyboard activities in system
- PRNG used a fixed value (seed) which may be produced by a TRNG. An adversary who knows the algorithm and seed can reproduce entire bit stream
- A PRF is similar to a PRNG, but produces a random value of fixed length (for example: a nonce or symmetric key)
- A PRF also uses some context specific values, for example: a used ID or and application ID

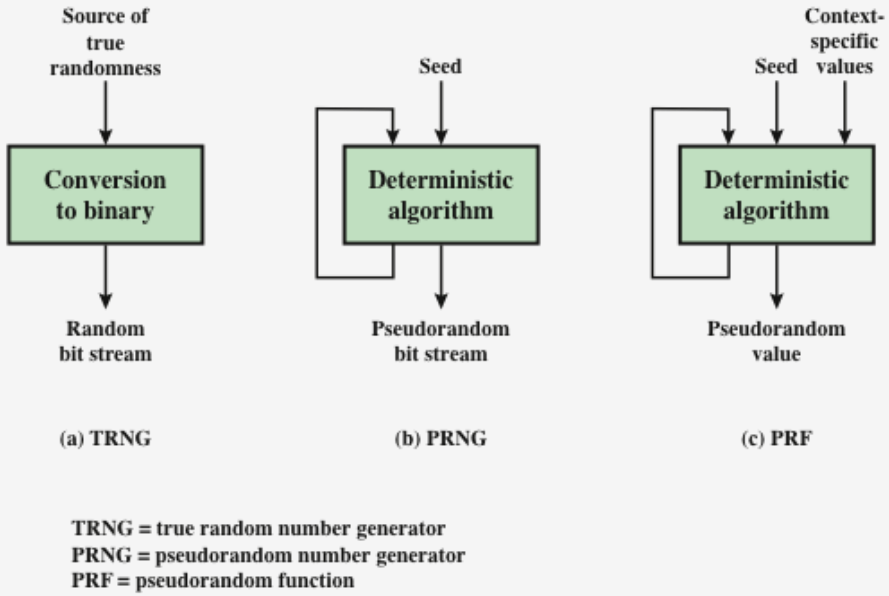


Figure 8.1 Random and Pseudorandom Number Generators

Stream Ciphers

- A key is input to a PRNG that produces a stream of 8-bit numbers that are apparently random
- Keystream is combined one byte at a time with the plaintext stream
- If using a TRNG we have a one-time-pad

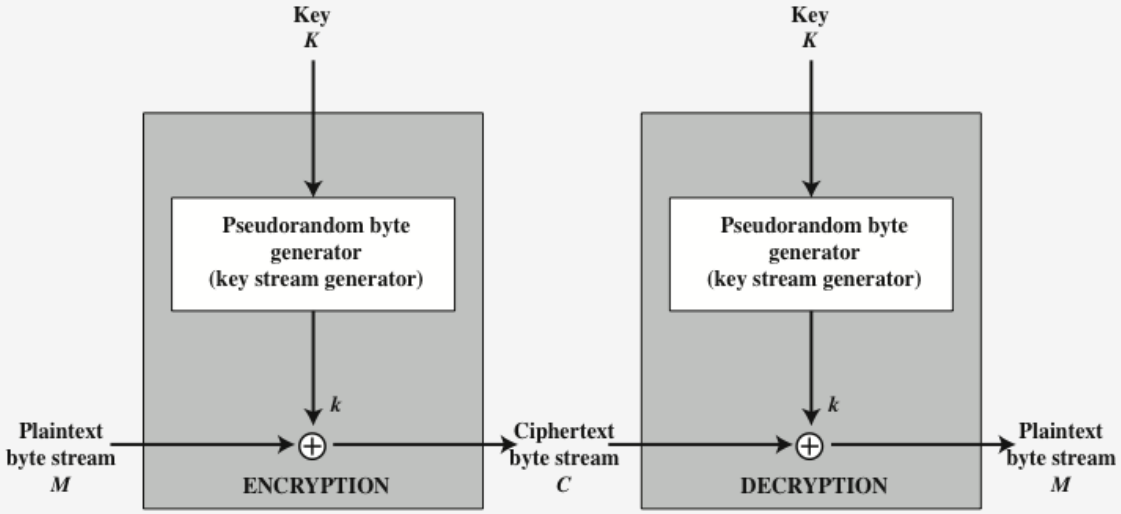


Figure 8.7 Stream Cipher Diagram

Stream Cipher Design Considerations

The encryption sequence should have a large period	• A pseudorandom number generator uses a function that produces a deterministic stream of bits that eventually repeats; the longer the period of repeat the more difficult it will be to do cryptanalysis
The keystream should approximate the properties of a true random number stream as close as possible	• There should be an approximately equal number of 1s and 0s • If the keystream is treated as a stream of bytes, then all of the 256 possible byte values should appear approximately equally often
A key length of at least 128 bits is desirable	• The output of the pseudorandom number generator is conditioned on the value of the input key • The same considerations that apply to block ciphers are valid
With a properly designed pseudorandom number generator a stream cipher can be as secure as a block cipher of comparable key length	• A potential advantage is that stream ciphers that do not use block ciphers as a building block are typically faster and use far less code than block ciphers

Stream vs. Block Ciphers

- Stream ciphers are typically faster and use far less code (less true with AES, which is quite efficient in software and hardware)
- One advantage of block ciphers is that we can reuse keys
- For applications requiring encryption/decryption of a stream of data (data communications channel, web link) a stream cipher may be the best alternative
- For applications dealing with blocks of data (file transfers, email, databases) block ciphers may be more appropriate
- But, either type of cipher can be used in virtually any application

Example stream cipher: RC4

- Designed in 1987 by Ron Rivest for RSA Security
- Variable key size stream cipher with byte-oriented operations
- Based on the use of a random permutation
- Eight to sixteen machine operations are required per output byte and the cipher can be expected to run very quickly in software
- Used in the Secure Sockets Layer/Transport Layer Security (SSL/TLS) standards that have been defined for communication between Web browsers and servers
- Is also used in the Wired Equivalent Privacy (WEP) protocol and the newer Wi-Fi Protected Access (WPA) protocol that are part of the IEEE 802.11 wireless LAN standard

RC4 Operation

- Vector S contains entries from 0 to 255 (initialized in that order)
- At all times, S contains a permutation of all numbers between 0 and 255
- Key K is transferred to temporary vector T
- Next we use T to produce initial permutation of S
- Next, stream generation involves swapping elements in S according to the current configuration of S
- To encrypt, XOR k with the next byte of plaintext
- To decrypt, XOR k with the next byte of ciphertext

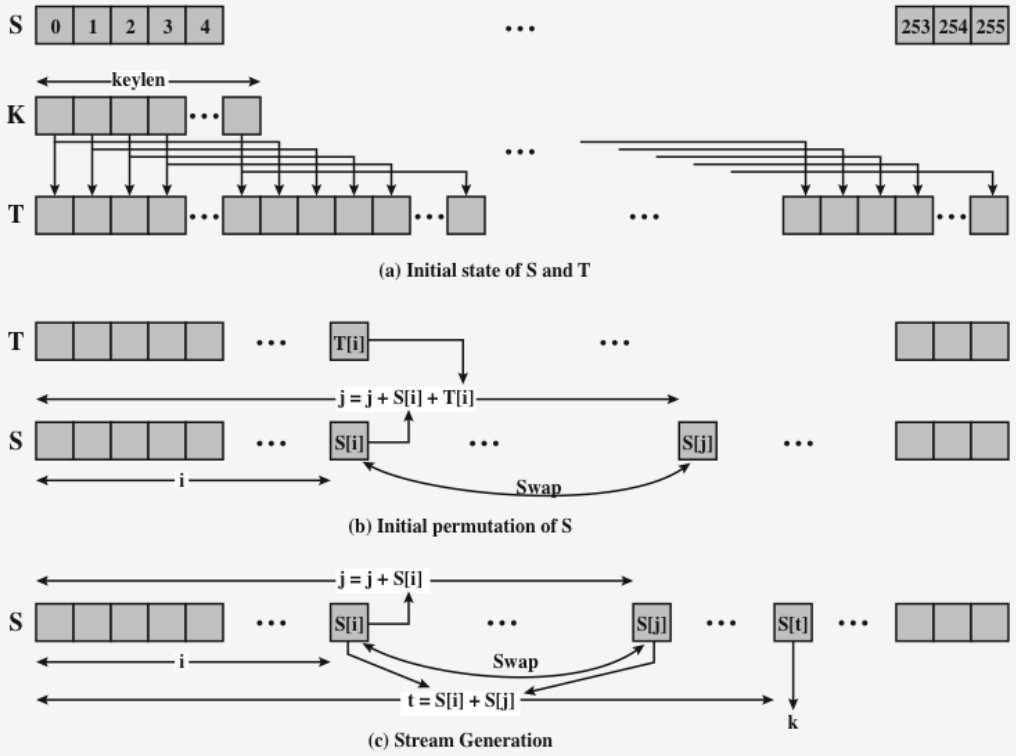


Figure 8.8 RC4

RC4 Operation (implementation)

1. Initialization

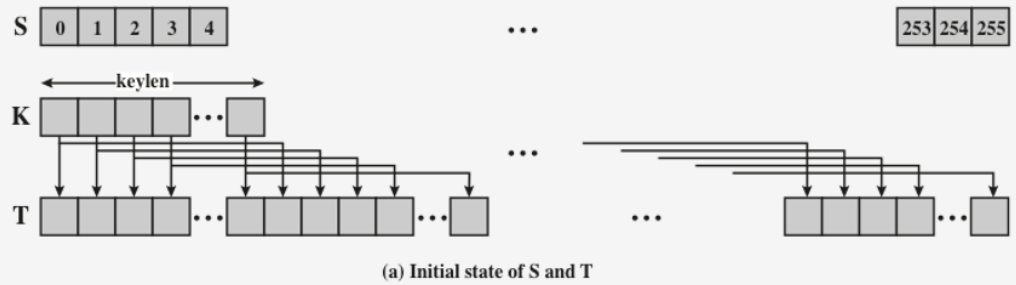
```
for i = 0 to 255 do
        S[i] = i;
        T[i] = K[i mod keylen]
```

2. Initial Permutation of S

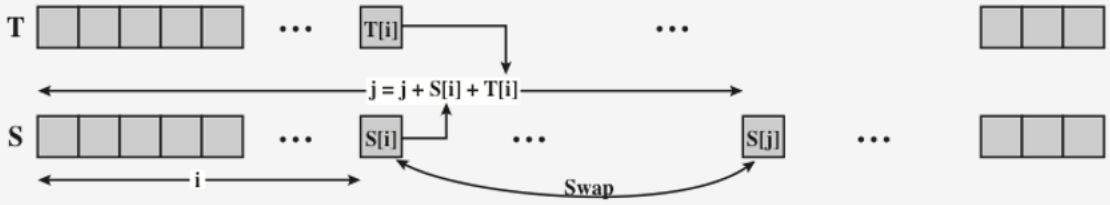
```
j = 0;
for i = 0 to 255 do
        j = (j+S[i]+T[i]) mod 256;
        Swap (S[i],S[j]);
```

3. Stream Generation

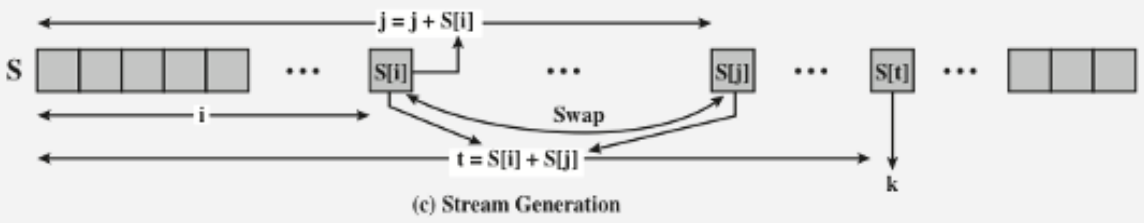
```
i , j = 0
while (true)
        i = (i+1) mod 256;
        j = (j+s[i]) mod 256;
        Swap (S[i],S[j]);
        t = (S[i]+S[j]) mod 256;
        k = S[t];
```



(a) Initial state of S and T



(b) Initial permutation of S



(c) Stream Generation

Review questions

- What is the differences between a block cipher and a stream cipher?

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

- What is the avalanche effect of an encryption algorithm?

The avalanche effect is a property of any encryption algorithm such that a small change in either the plaintext or the key produces a significant change in the ciphertext

- What is the difference between diffusion and confusion?

In diffusion, the statistical structure of the plaintext is dissipated in the ciphertext, by having each plaintext digit affect the value of many ciphertext digits. Confusion seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible. Thus, even if the attacker can get some handle on the statistics of the ciphertext, the way in which the key was used to produce that ciphertext is so complex as to make it difficult to deduce the key. This is achieved by the use of a complex substitution algorithm.

Summary

Fundamental concepts

Classical Encryption Techniques

- ✓Symmetric Cipher Model
- ✓Substitution Techniques
- ✓Transposition Techniques
- ✓Rotor Machines
- ✓Steganography

Feistel cipher structure and design

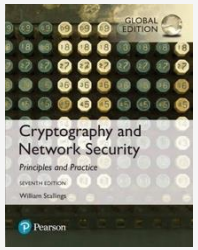
Symmetric block encryption algorithms

- ✓ DES
- ✓ 3DES
- ✓ AES

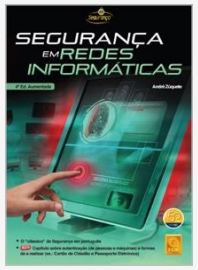
Stream ciphers: RC4

Bibliography

Cryptography and network security, Stallings, Pearson, 2017, Chapter 3: Classical Encryption Techniques, Chapter 4: Block Ciphers and the Data Encryption Standard, Chapter 8: Stream Ciphers



Segurança em Redes Informáticas, Capítulo 2: Criptografia



Segurança Prática em Sistemas e Redes com Linux, Capítulo 1: Conceitos fundamentais