Week 3

Additional Reading for Week 3:

These slides are taken directly from Chapter 7 of the textbook by William Stallings.

You are required to study for this for next week’s workshop.

Chapter 7

Block Cipher Operation

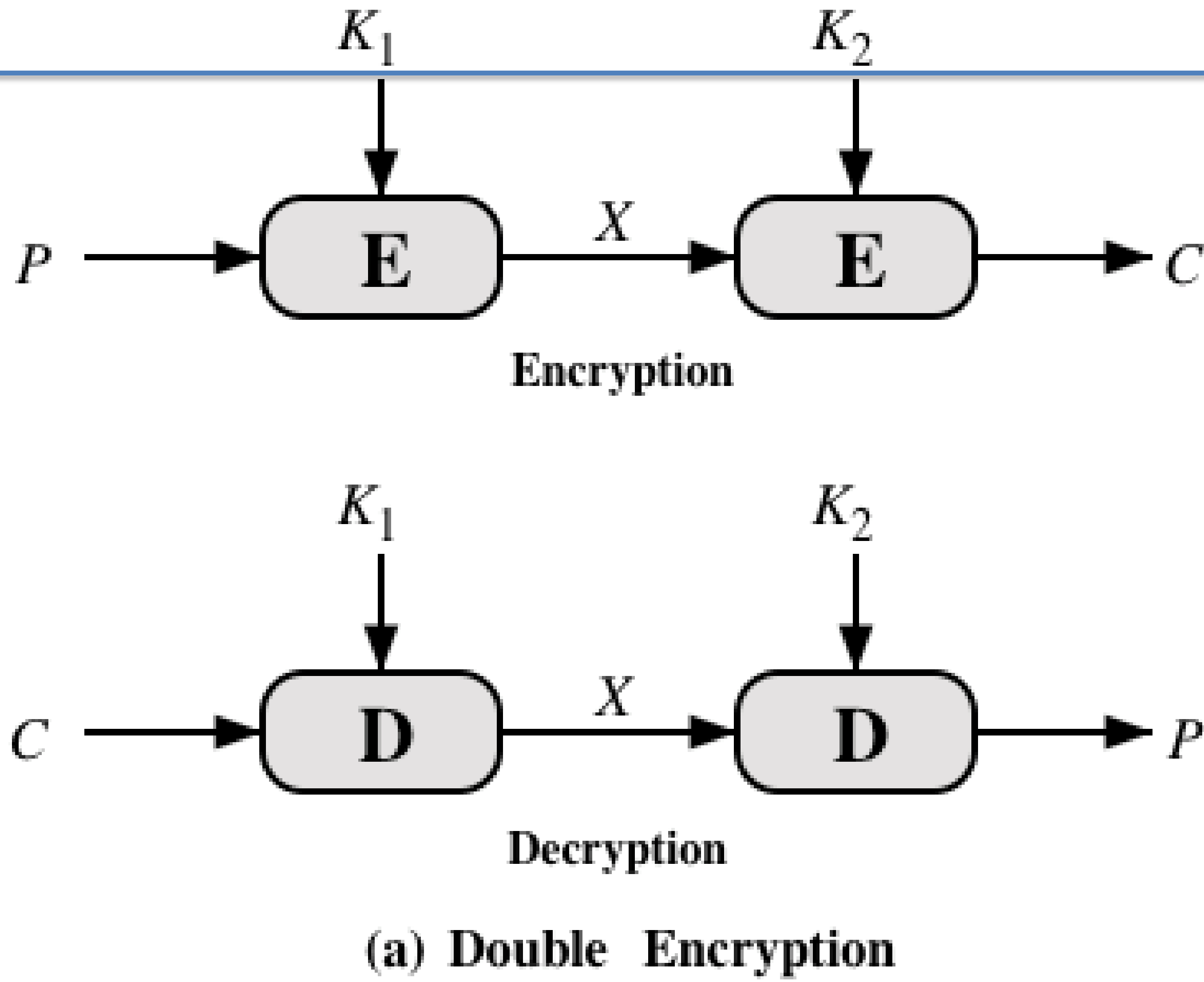


Figure 7.1 Multiple Encryption

Meet-in-the-Middle Attack

The use of double DES results in a mapping that is not equivalent to a single DES encryption

The meet-in-the-middle attack algorithm will attack this scheme and does not depend on any particular property of DES but will work against any block encryption cipher

Triple-DES with Two-Keys

- Obvious counter to the meet-in-the-middle attack is to use three stages of encryption with three different keys
  - This raises the cost of the meet-in-the-middle attack to 2^{112}, which is beyond what is practical
  - Has the drawback of requiring a key length of 56 x 3 = 168 bits, which may be somewhat unwieldy
  - As an alternative Tuchman proposed a triple encryption method that uses only two keys
- 3DES with two keys is a relatively popular alternative to DES and has been adopted for use in the key management standards ANSI X9.17 and ISO 8732

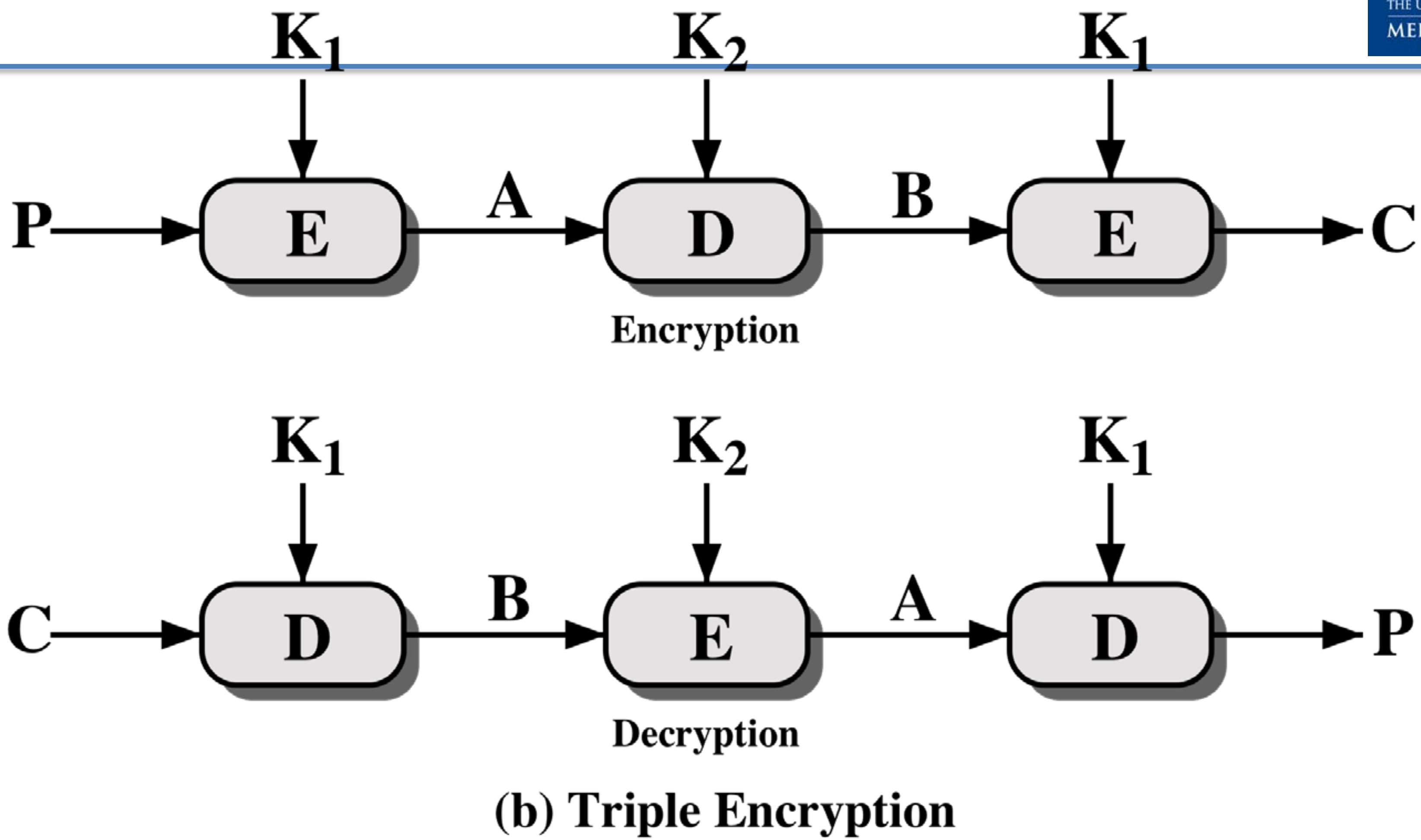


Figure 7.1 Multiple Encryption

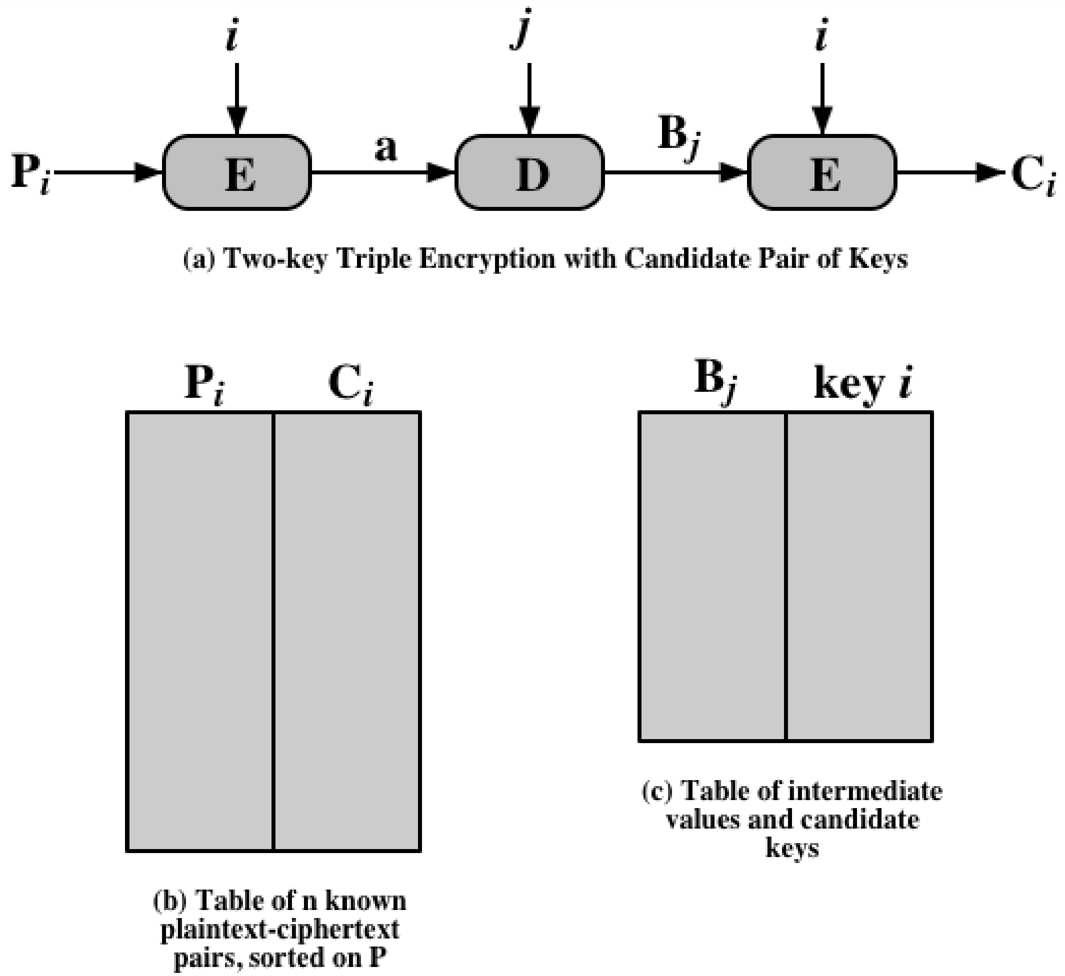


Figure 7.2 Known-Plaintext Attack on Triple DES

THE UNIVERSITY OF MELBOURNE

Triple DES with Three Keys

- Many researchers now feel that three-key 3DES is the preferred alternative

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 ve adopted three-key 3DES including PGP and S/MIME

Modes of Operation

- A technique for enhancing the effect of a cryptographic algorithm or adapting the algorithm for an application
- To apply a block cipher in a variety of applications, five *modes of operation* have been defined by NIST
  - The five modes are intended to cover a wide variety of applications of encryption for which a block cipher could be used
  - These modes are intended for use with any symmetric block cipher, including triple DES and AES

Table 7.1 Block Cipher Modes of Operation

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

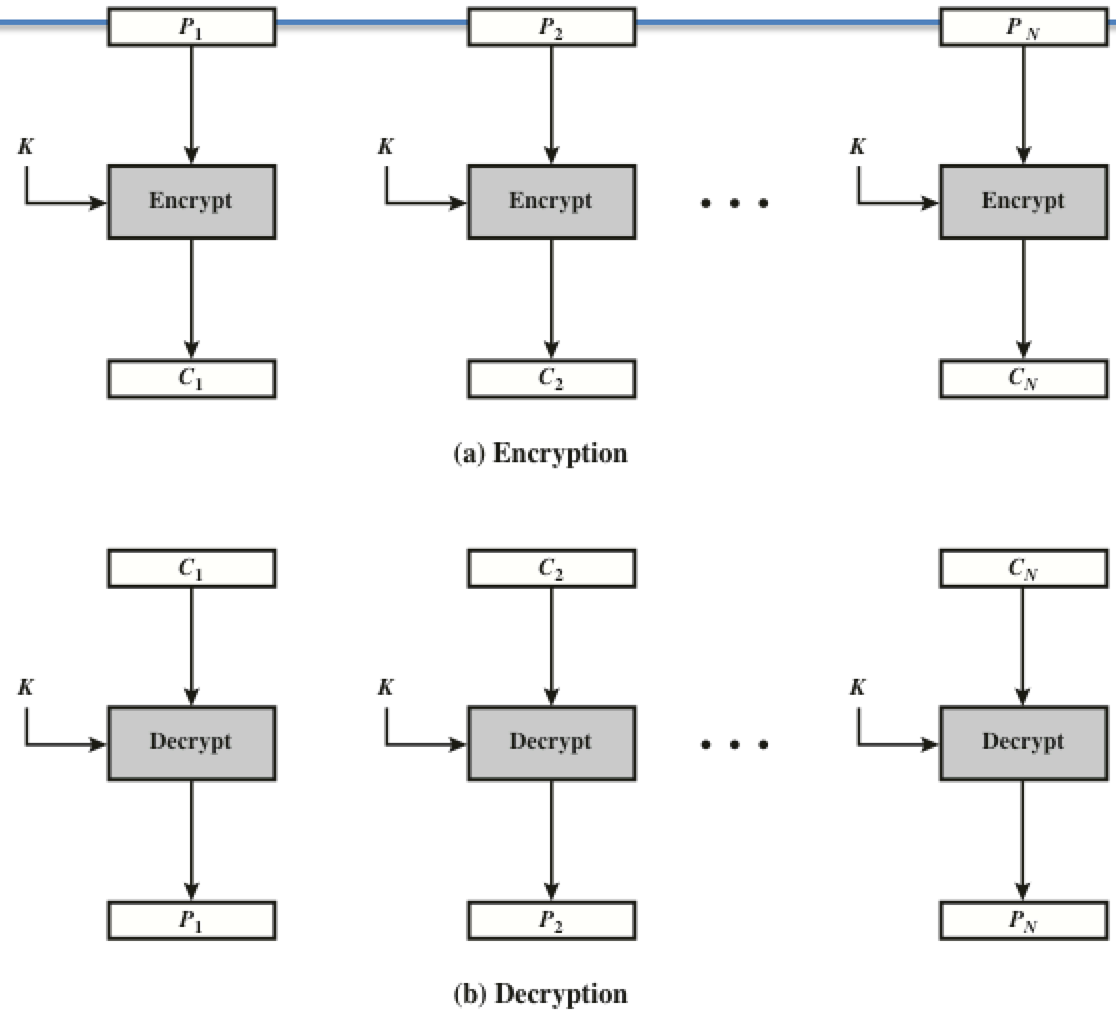


Figure 7.3 Electronic Codebook (ECB) Mode

THE UNIVERSITY OF MELBOURNE

Criteria and properties for evaluating and constructing block cipher modes of operation that are superior to ECB:

- Overhead
- Error recovery
- Error propagation
- Diffusion
- Security

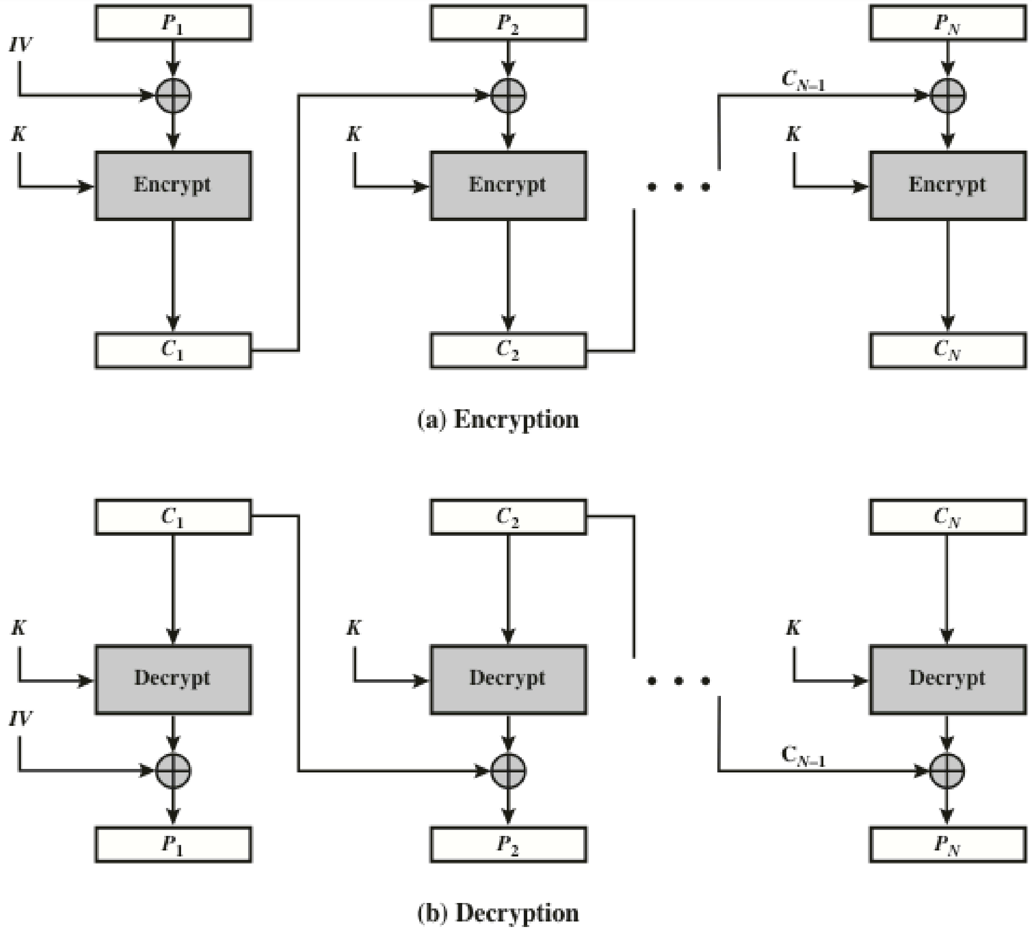


Figure 7.4 Cipher Block Chaining (CBC) Mode

Cipher Feedback Mode

- For AES, DES, or any block cipher, encryption is performed on a block of *b* bits
  - In the case of DES *b* = 64
  - In the case of AES *b* = 128

There are three modes that make it possible to convert a block cipher into a stream cipher:

- Cipher feedback (CFB) mode
- Output feedback (OFB) mode
- Counter (CTR) mode

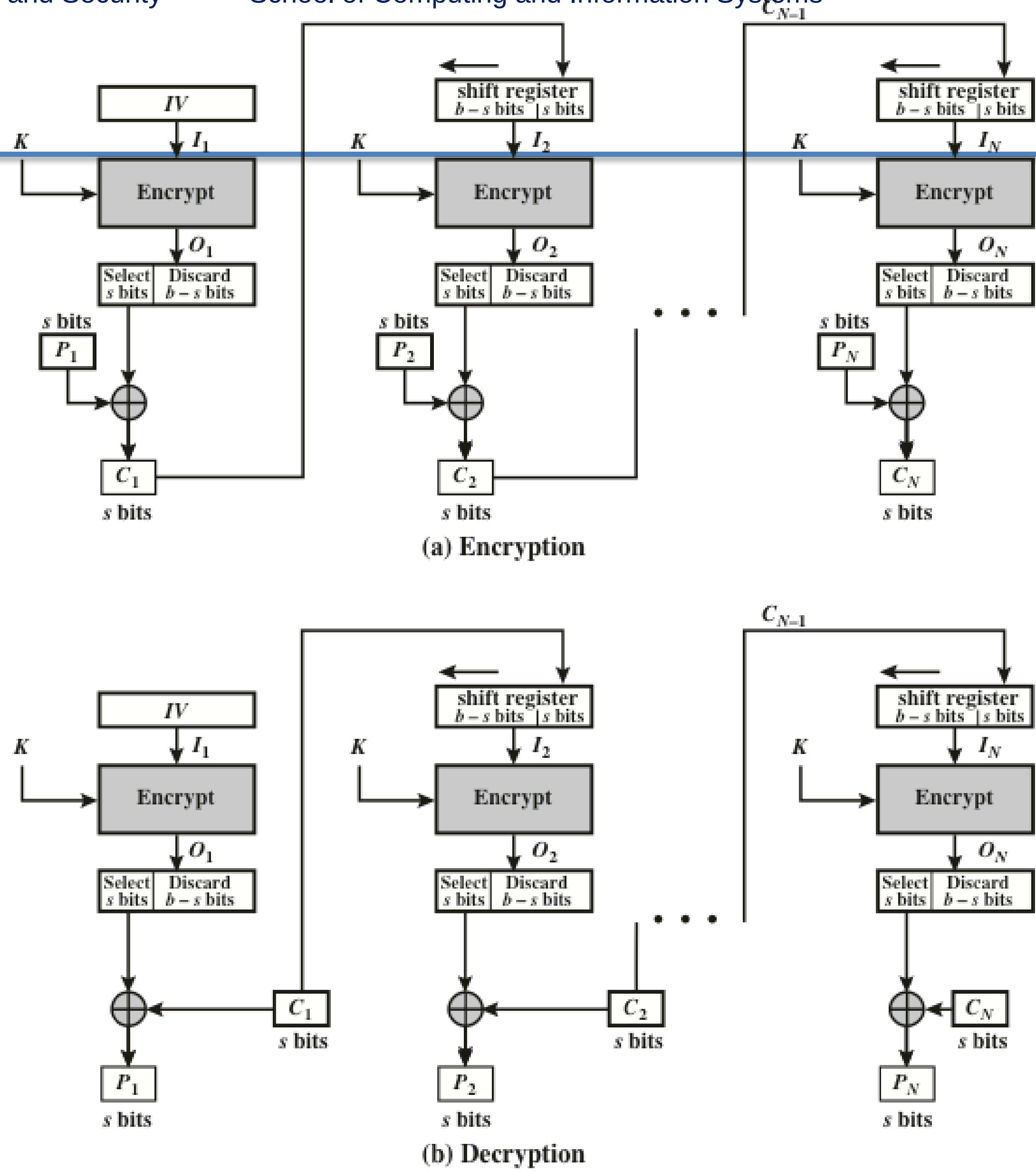
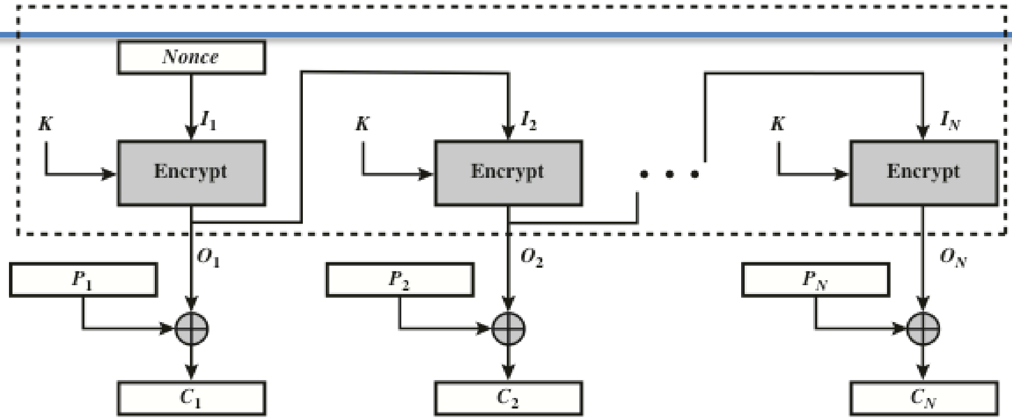
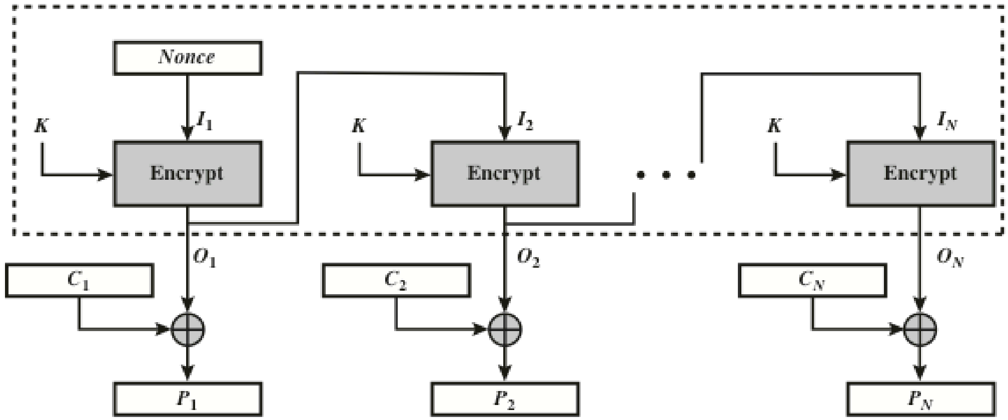


Figure 7.5 s-bit Cipher Feedback (CFB) Mode



(a) Encryption



(b) Decryption

Figure 7.6 Output Feedback (OFB) Mode

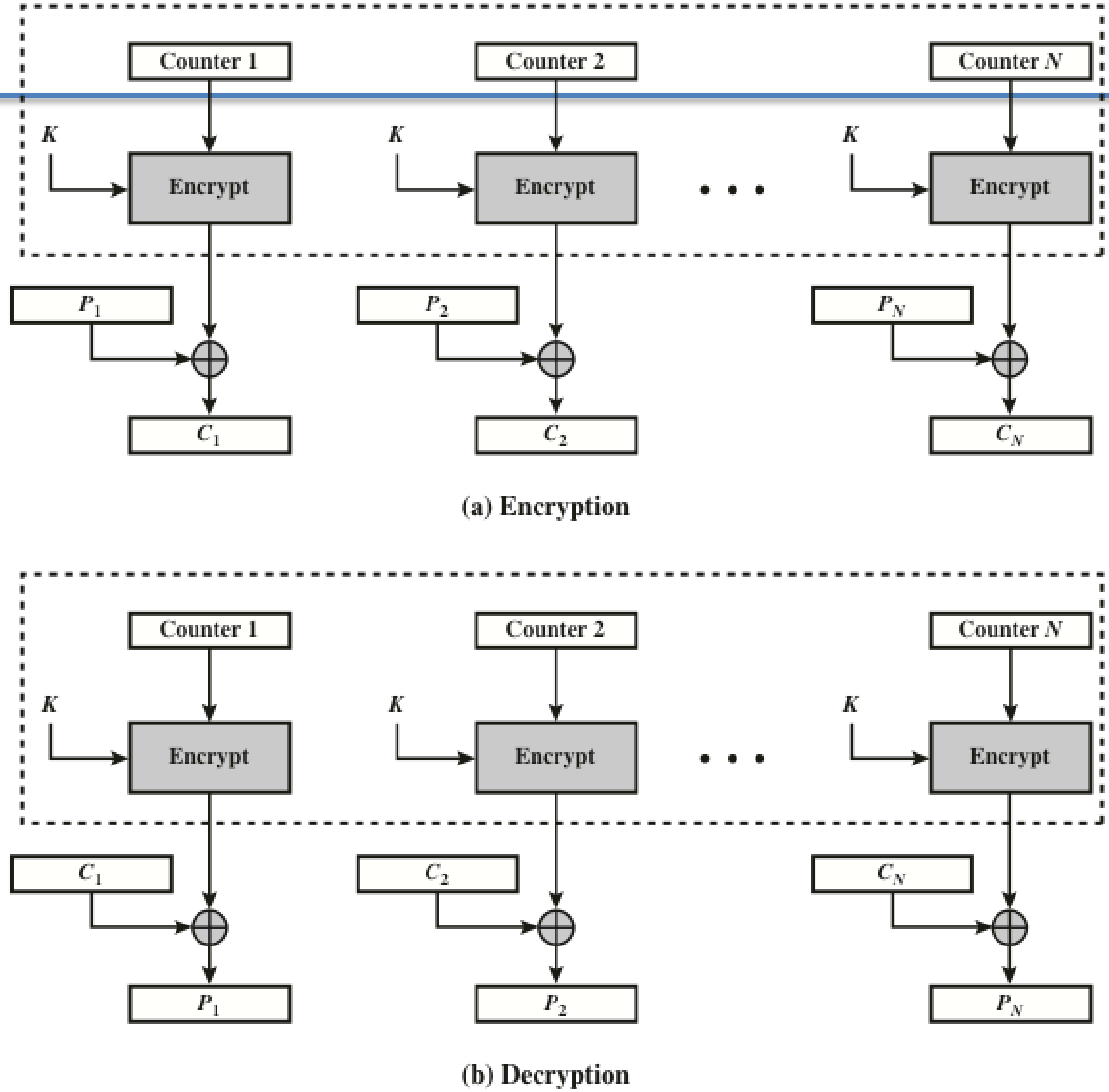
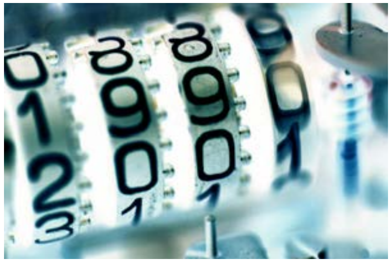


Figure 7.7 Counter (CTR) Mode

Advantages of CTR



- Hardware efficiency
- Software efficiency
- Preprocessing
- Random access
- Provable security
- Simplicity

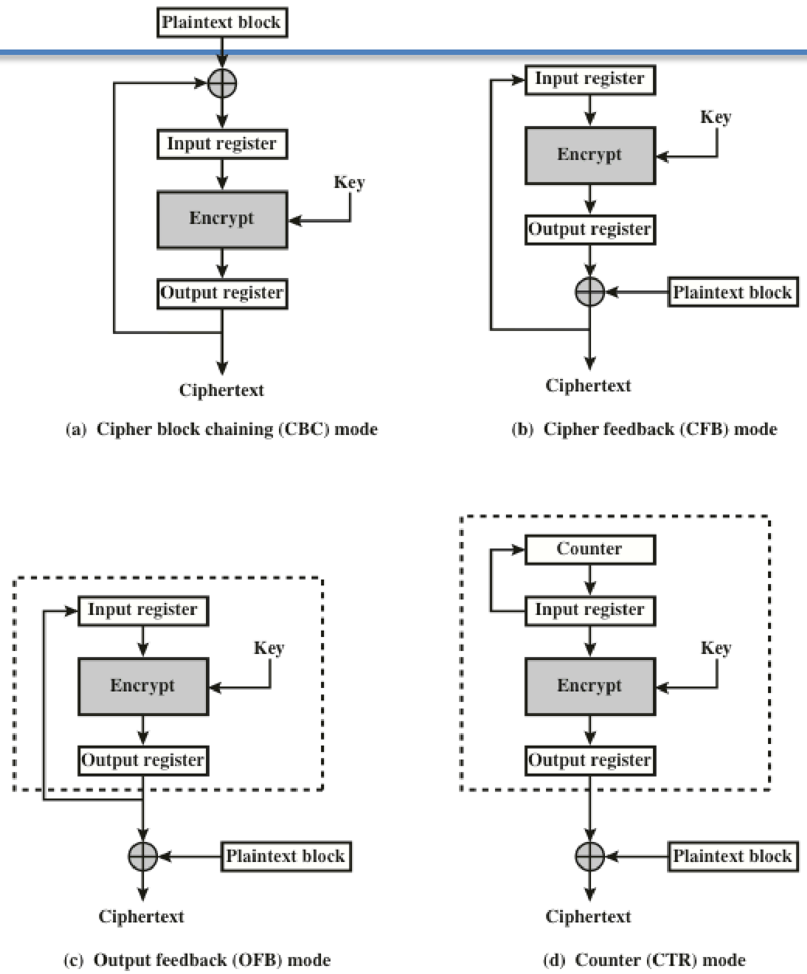


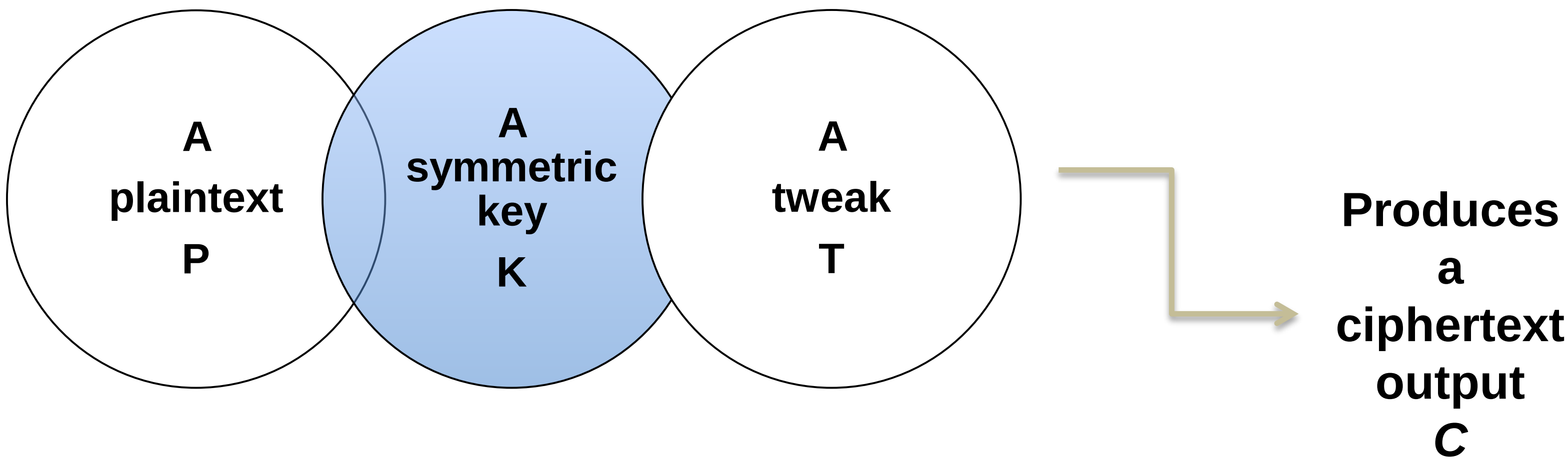
Figure 7.8 Feedback Characteristic of Modes of Operation

XTS-AES Mode for Block-Oriented Storage Devices

- Approved as an additional block cipher mode of operation by NIST in 2010
- Mode is also an IEEE Standard, IEEE Std 1619-2007
  - Standard describes a method of encryption for data stored in sector-based devices where the threat model includes possible access to stored data by the adversary
  - Has received widespread industry support

Tweakable Block Ciphers

- XTS-AES mode is based on the concept of a *tweakable block cipher*
- General structure:
  - Has three inputs:



- Tweak need not be kept secret
  - Purpose is to provide variability

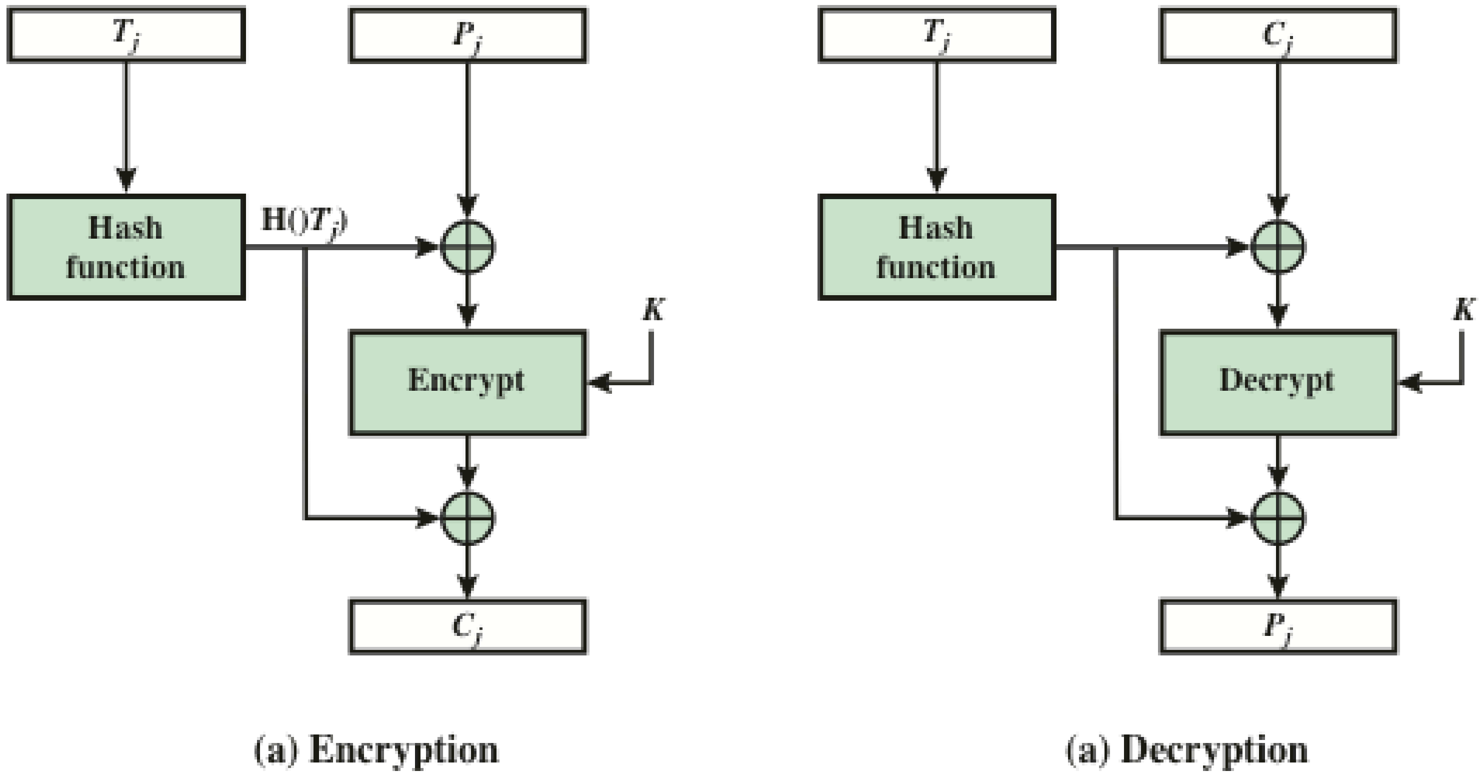


Figure 7.9 Tweakable Block Cipher

Storage Encryption Requirements

- The requirements for encrypting stored data, also referred to as "data at rest", differ somewhat from those for transmitted data
- The P1619 standard was designed to have the following characteristics:
  - The ciphertext is freely available for an attacker
  - The data layout is not changed on the storage medium and in transit
  - Data are accessed in fixed sized blocks, independently from each other
  - Encryption is performed in 16-byte blocks, independently from each other
  - There are no other metadata used, except the location of the data blocks within the whole data set
  - The same plaintext is encrypted to different ciphertexts at different locations, but always to the same ciphertext when written to the same location again
  - A standard conformant device can be constructed for decryption of data encrypted by another standard conformant device

XTS-AES Operation on Single Block

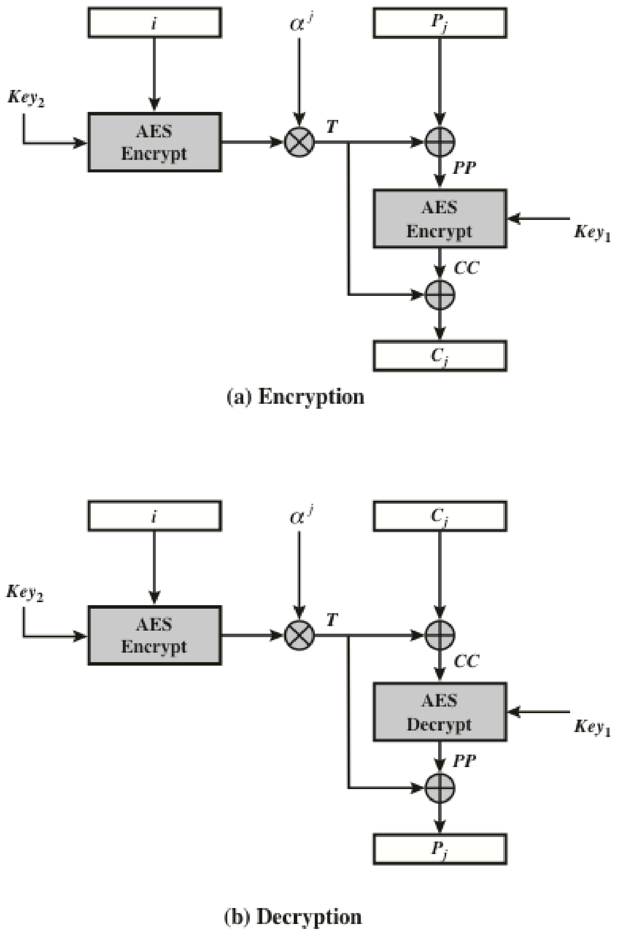


Figure 7.10 XTS-AES Operation on Single Block

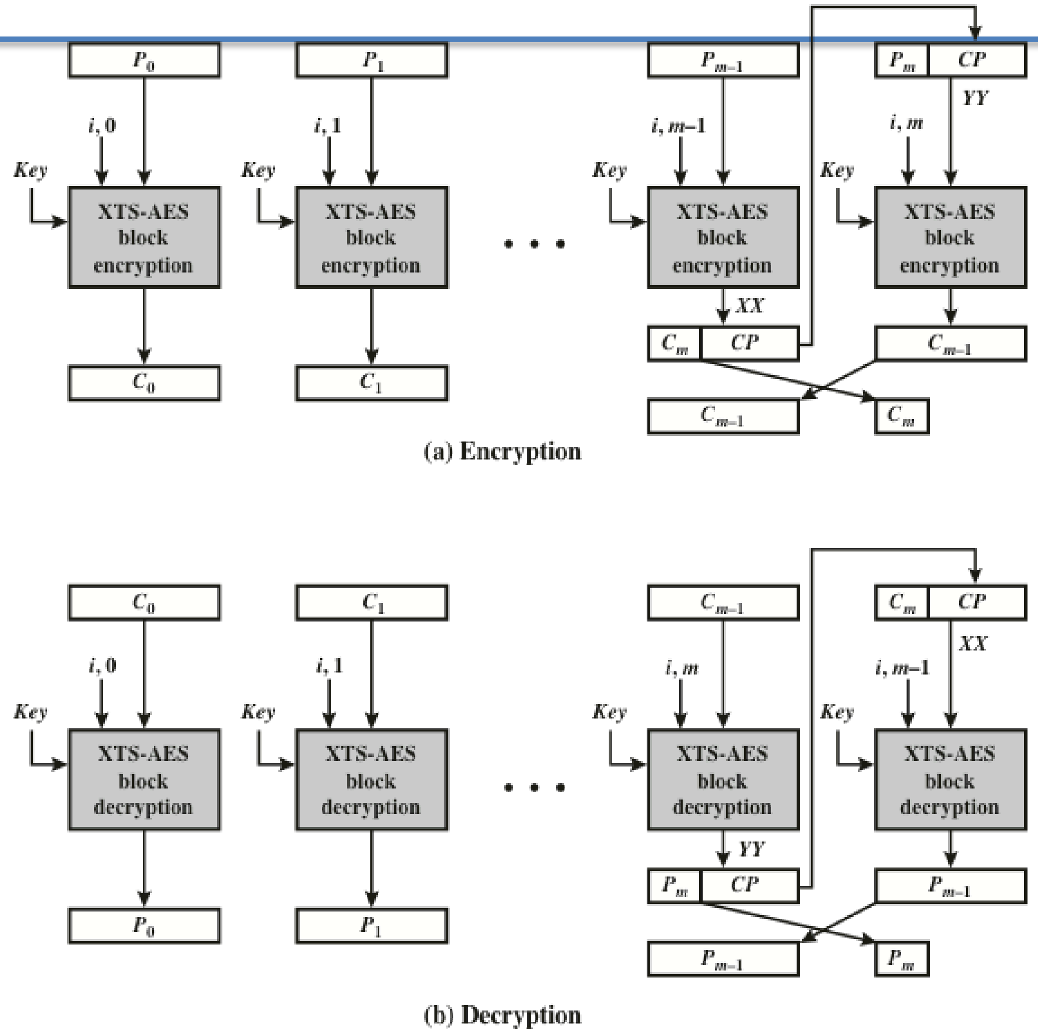


Figure 7.11 XTS-AES Mode

Format-Preserving Encryption (FPE)

- Refers to any encryption technique that takes a plaintext in a given format and produces a ciphertext in the same format
  - For example: credit cards consist of 16 decimal digits. An FPE that can accept this type of input would produce a ciphertext output of 16 decimal digits. (Note that the ciphertext need not be, and in fact in unlikely to be, a valid credit card number.) But it will have the same format and can be stored in the same way as credit card number plaintext.

Table 7.2

Comparison of Format-Preserving Encryption and AES

	Credit Card	Tax ID	Bank Account Number
Plaintext	8123 4512 3456 6780	219-09-9999	800N2982K-22
FPE	8123 4521 7292 6780	078-05-1120	709G9242H-35
AES (hex)	af411326466add24 c86abd8aa525db7a	7b9af4f3f218ab25 07c7376869313afa	9720ec7f793096ff d37141242e1c51bd

THE UNIVERSITY OF MELBOURNE

Motivation

FPE facilitates the retrofitting of encryption technology to legacy applications, where a conventional encryption mode might not be feasible because it would disrupt data fields/pathways

FPE has emerged as a useful cryptographic tool, whose applications include financial-information security, data sanitization, and transparent encryption of fields in legacy databases

The principal benefit of FPE is that it enables protection of particular data elements, while still enabling workflows that were in place before FPE was in use

- No database schema changes and minimal application changes are required
- Only applications that need to see the plaintext of a data element need to be modified and generally these modifications will be minimal

Some examples of legacy applications where FPE is desirable are:

- COBOL data-processing applications
- Database applications
- FPE-encrypted characters can be significantly compressed for efficient transmission

Difficulties in Designing an FPE

- A general-purpose standardized FPE should meet a number of requirements:
  - The ciphertext is of the same length and format as the plaintext
  - It should be adaptable to work with a variety of character and number types
  - It should work with variable plaintext length
  - Security strength should be comparable to that achieved with AES
  - Security should be strong even for very small plaintext lengths

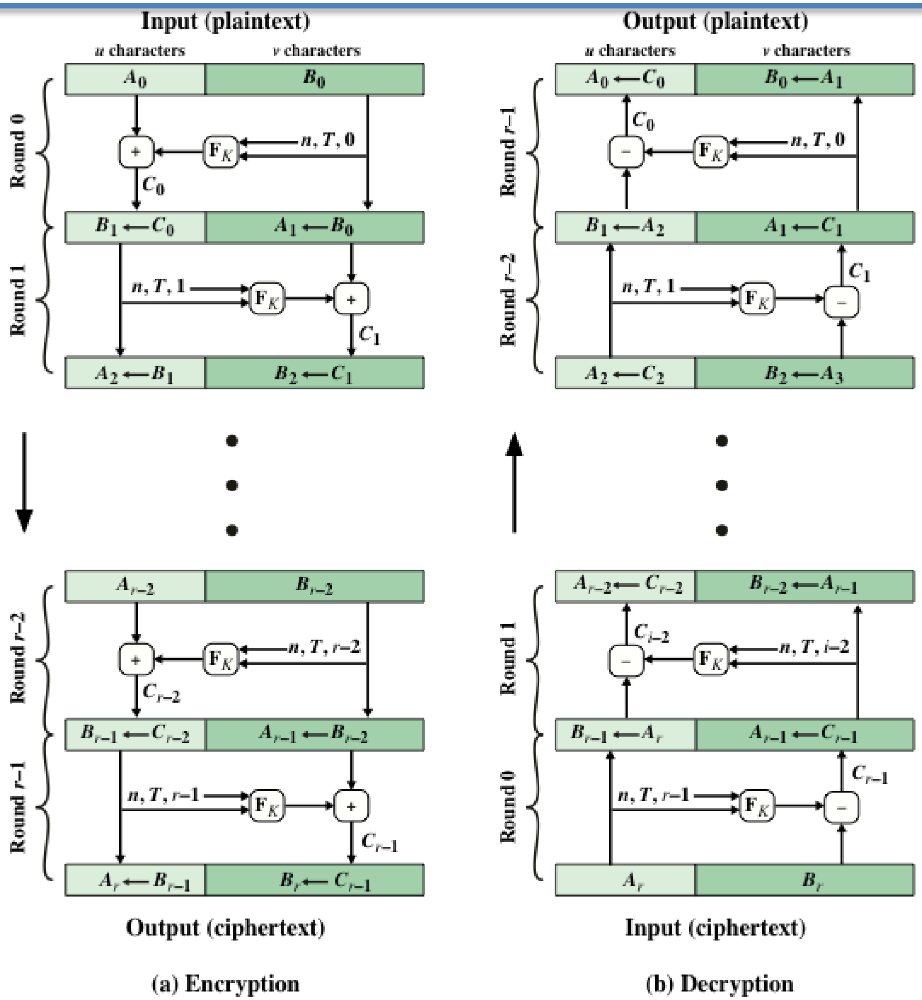


Figure 7.12 Feistel Structure for Format-Preserving Encryption

Character Strings

- The NIST, and the other FPE algorithms that have been proposed, are used with plaintext consisting of a string of elements, called characters
- A finite set of two or more symbols is called an *alphabet*
- The elements of an alphabet are called *characters*
- A *character string* is a finite sequence of characters from an alphabet
- Individual characters may repeat in the string
- The number of different characters in an alphabet is called the *base* (also referred to as the *radix)* of the alphabet

Table 7.3 Notation and Parameters Used in FPE Algorithms

(a) Notation

$[x]^s$	Converts an integer into a byte string; it is the string of s bytes that encodes the number x, with $0 \le x < 2^{8s}$. The equivalent notation is $\text{STR}_2^{8s}(x)$.
LEN(X)	Length of the character string X.
$\text{NUM}_{radix}(X)$	Converts strings to numbers. The number that the numeral string X represents in base *radix*, with the most significant character first. In other words, it is the non-negative integer less than $radix^{\text{LEN}(X)}$ whose most-significant-character-first representation in base *radix* is X.
$\text{PRF}_K(X)$	A pseudorandom function that produces a 128-bit output with X as the input, using encryption key K.
$\text{STR}_{radix}^m(x)$	Given a nonnegative integer x less than $radix^m$, this function produces a representation of x as a string of m characters in base *radix*, with the most significant character first.
$[i .. j]$	The set of integers between two integers i and j, including i and j.
$X[i .. j]$	The substring of characters of a string X from $X[i]$ to $X[j]$, including $X[i]$ and $X[j]$.
REV(X)	Given a bit string, X, the string that consists of the bits of X in reverse order.

Table 7.3

Notation and Parameters Used in FPE Algorithms

(b) Parameters

radix	The base, or number of characters, in a given plaintext alphabet.
tweak	Input parameter to the encryption and decryption functions whose confidentiality is not protected by the mode.
tweakradix	The base for tweak strings
minlen	Minimum message length, in characters.
maxlen	Maximum message length, in characters.
maxTlen	Maximum tweak length

t

Prerequisites:
Approved, 128-bit block cipher, CIPH;
Key, K, for the block cipher;

Input:
Nonempty bit string, X, such that LEN(X) = is a multiple of 128.
Output:
128-bit block, Y

Steps:
1. Let m = LEN(X)/128.
2. Partition X into m 128-bit blocks $X_1, \ldots, X_m$, so that $X = X_1 \| \ldots \| X_m$
3. Let $Y_0 = [0]^{16}$
4. For j from 1 to m:
4.i let $Y_j = \mathrm{CIPH}_K(Y_{j-1} \oplus X_j)$.
6. Return Y_m.

Figure 7.13 Algorithm PRF(*X*)

Prerequisites:
Approved, 128-bit block cipher, CIPH;
Key, K, for the block cipher;
Base, *radix*, for the character alphabet;
Range of supported message lengths, [minlen .. maxlen];
Maximum byte length for tweaks, maxTlen.

Inputs:
Character string, X, in base radix of length n such that $n \in$ [minlen .. maxlen];
Tweak T, a byte string of byte length t, such that $t \in$ [0 .. maxTlen].
Output:
Character string, Y, such that LEN(Y) = n.

Steps:
1. Let $u = \lfloor n/2 \rfloor$; $v = n - u$.
2. Let $A = X[1 .. u]$; $B = X[u + 1 .. n]$.
3. Let $b = \lceil \lceil v \, \mathrm{LOG}_2(radix) \rceil / 8 \rceil$; $d = 4 \lceil b/4 \rceil + 4$
4. Let $P = [1]^1 \,\|\, [2]^1 \,\|\, [1]^1 \,\|\, [radix]^3 \,\|\, [10]^1 \,\|\, [u \bmod 256]^1 \,\|\, [n]^4 \,\|\, [t]^4$.
5. For i from 0 to 9:
   i. Let $Q = T \,\|\, [0]^{(-t-b-1) \bmod 16} \,\|\, [i]^1 \,\|\, [\mathrm{NUM}_{radix}(B)]^b$.
   ii. Let $R = \mathrm{PRF}_K(P \,\|\, Q)$.
   iii. Let S be the first d bytes of the following string of $\lceil d/16 \rceil$ 128-bit blocks:
   $R \,\|\, \mathrm{CIPH}_K(R \oplus [1]^{16}) \,\|\, \mathrm{CIPH}_K(R \oplus [2]^{16}) \,\|\, \ldots \|\, \mathrm{CIPH}_K(R \oplus [\lceil d/16 \rceil - 1]^{16})$.
   iv. Let $y = \mathrm{NUM}_2(S)$.
   v. If i is even, let $m = u$; else, let $m = v$.
   vi Let $c = (\mathrm{NUM}_{radix}(A) + y) \bmod radix^m$.
   vii. Let $C = \mathrm{STR}^m_{radix}(c)$.
   viii. Let $A = B$.
   ix. Let $B = C$.
6. Return $Y = A \,\|\, B$.

Figure 7.14 Algorithm FF1 (FFX[Radix])

Approved, 128-bit block cipher, CIPH;
Key, K, for the block cipher;
Base, *tweakradix*, for the tweak character alphabet;
Range of supported message lengths, [*minlen* .. *maxlen*]
Maximum supported tweak length, *maxTlen*.

Inputs:
Numeral string, X, in base *radix*, of length n such that $n \in$ [*minlen* .. *maxlen*];
Tweak numeral string, T, in base *tweakradix*, of length t such that $t \in$ [0 .. *maxTlen*].
Output:
Numeral string, Y, such that LEN(Y) = n.

Steps:
1. Let $u = \lfloor n/2 \rfloor$; $v = n - u$.
2. Let $A = X[1 .. u]$; $B = X[u + 1 .. n]$.
3. If $t>0$, $P = [radix]^1 \,||\, [t]^1 \,||\, [n]^1 \,||\, [\mathrm{NUM}_{tweakradix}(T)]^{13}$;
   else $P = [radix]^1 \,||\, [0]^1 \,||\, [n]^1 \,||\, [0]^{13}$.
4. Let $J = \mathrm{CIPH}_K(P)$
5. For i from 0 to 9:
   i. Let $Q \leftarrow [i]^1 \,||\, [\mathrm{NUM}_{radix}(B)]^{15}$
   ii. Let $Y \leftarrow \mathrm{CIPH}_J(Q)$.
   iii. Let $y \leftarrow \mathrm{NUM}_2(Y)$.
   iv. If i is even, let $m = u$; else, let $m = v$.
   v. Let $c = (\mathrm{NUM}_{radix}(A) + y) \bmod radix^m$.
   vi. Let $C = \mathrm{STR}^m_{radix}(c)$.
   vii. Let $A = B$.
   viii. Let $B = C$.
6. Return $Y = A \,||\, B$.

Figure 7.15 Algorithm FF2 (VAES3)

Approved, 128-bit block cipher, CIPH;
Key, K, for the block cipher;
Base, $radix$, for the character alphabet such that $radix \in [2 .. 2^{16}]$;
Range of supported message lengths, $[minlen .. maxlen]$,
such that $minlen \geq 2$ and $maxlen \leq 2\lfloor \log_{radix}(2^{96}) \rfloor$.

Inputs:
Numeral string, X, in base $radix$ of length n such that $n \in [minlen .. maxlen]$;
Tweak bit string, T, such that LEN(T) = 64.
Output:
Numeral string, Y, such that LEN(Y) = n.

Steps:
1. Let $u = \lceil n/2 \rceil$; $v = n - u$.
2. Let $A = X[1 .. u]$; $B = X[u + 1 .. n]$.
3. Let $T_L = T[0..31]$ and $T_R = T[32..63]$
4. For i from 0 to 7:
   i. If i is even, let $m = u$ and $W = T_R$, else let $m = v$ and $W = T_L$.
   ii. Let $P = \mathrm{REV}([\mathrm{NUM}_{radix}(\mathrm{REV}(B))]^{12}) \,||\, [W \oplus \mathrm{REV}([i]^4])$.
   iii. Let $Y = \mathrm{CIPH}_K(P)$.
   iv. Let $y = \mathrm{NUM}_2(\mathrm{REV}(Y))$.
   v. Let $c = (\mathrm{NUM}_{radix}(\mathrm{REV}(A)) + y) \bmod radix^m$.
   vi. Let $C = \mathrm{REV}(\mathrm{STR}^m_{radix}(c))$.
   vii. Let $A = B$.
   viii. Let $B = C$.
5. Return $A \,||\, B$.

Figure 7.16 Algorithm FF3 (BPS-BC)

Summary

- Multiple encryption and triple DES
  - Double DES
  - Triple DES with two keys
  - Triple DES with three keys
- Electronic codebook
- Cipher block chaining mode
- Format-preserving encryption
  - Motivation
  - Difficulties in designing
  - Feistel structure
  - NIST methods

- Cipher feedback mode
- Output feedback mode
- Counter mode
- XTS-AES mode for block-oriented storage devices
  - Tweakable block ciphers
  - Storage encryption requirements
  - Operation on a single block
  - Operation on a sector