Multiple Ciphers and Modes of Operation

Yih-Kuen Tsay

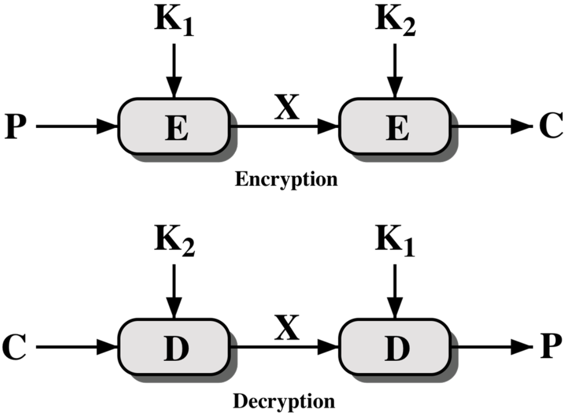
Department of Information Management
National Taiwan University

Bettering DES

Given the vulnerability of DES to a brute-force attack, there had been (before AES) considerable interest in finding an alternative:

- Completely new algorithms: Blowfish, RC5, ...
- Multiple encryption with DES and multiple keys (to preserve the existing investment in software and equipment):
  - Double DES
  - Triple DES

Multiple Encryption: Double DES



(a) Double Encryption

Source: Figure 6.1, Stallings 2014

Reduction to a Single Stage?

- Question: Given any two keys K_1 and K_2, would it be possible to find a key K_3 such that

$$E_{K_2}(E_{K_1}(P)) = E_{K_3}(P)?$$

- If so, then any multiple encryption would be equivalent to some single encryption.
- But, this is unlikely. (Affirmed in 1992.)
  - There are $2^{64}! > 10^{10^{20}}$ distinct permutations of the set of 2^{64} different 64-bit blocks.
  - Each 56-bit DES key defines one such permutation; $2^{56} < 10^{17}$.

Meet-in-the-Middle Attack

If we have $C = E_{K_2}(E_{K_1}(P))$, then for some X,

$$E_{K_1}(P) = X = D_{K_2}(C)$$

Given a known pair (P, C), the meet-in-the-middle attack proceeds as follows:

1. Encrypt P for all 2^{56} possible values of K_1 and then sort and store the results in a table.
2. Decrypt C using each possible value of K_2 and check the result against the table.
3. If a match occurs, then test the two keys against a new known pair.

Multiple Encryption: Triple DES

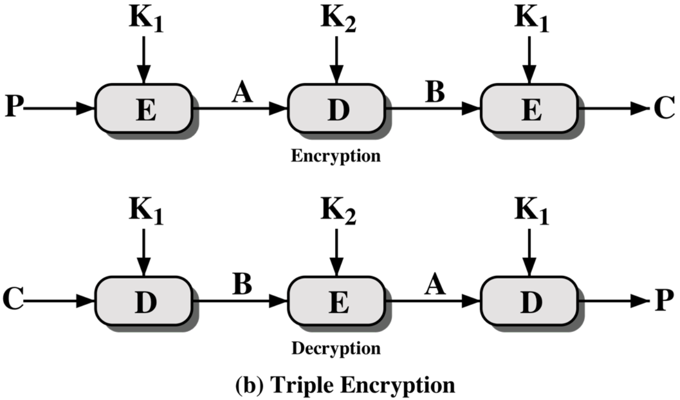


Figure 6.1 Multiple Encryption

Source: Figure 6.1, Stallings 2014

Two-Key Triple DES

- Proposed by Tuchman
- Encryption: $C = E_{K_1}(D_{K_2}(E_{K_1}(P)))$
- Interoperable with DES:

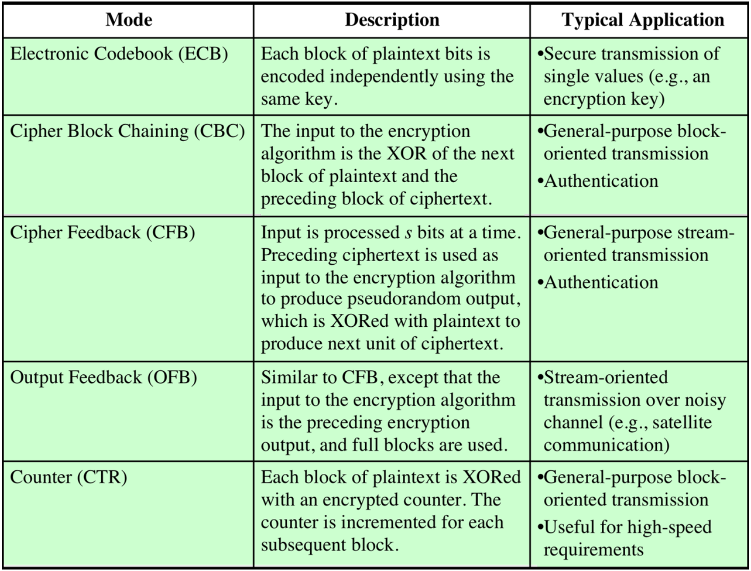
$$E_{K_1}(D_{K_1}(E_{K_1}(P))) = E_{K_1}(P)$$

- Adopted in ANS X9.17, ISO 8732, etc.
- No known practical cryptanalytic attacks

Three-Key Triple DES

- Many researchers now prefer three-key triple DES
- Encryption: $C = E_{K_3}(D_{K_2}(E_{K_1}(P)))$
- Backward compatible with DES by setting $K_3 = K_2$ or $K_2 = K_1$
- Adopted in PGP, S/MIME, etc.

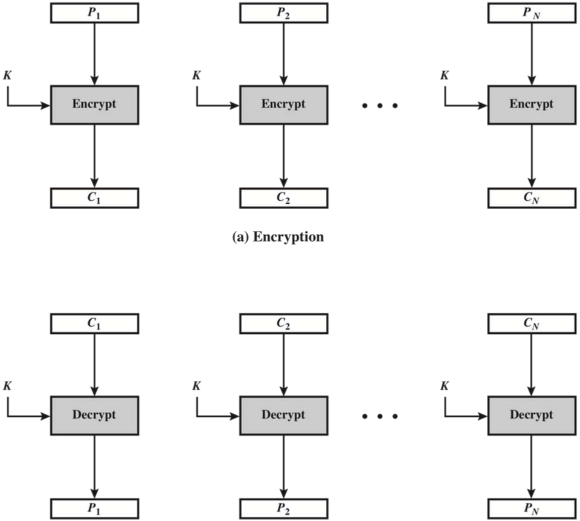
Modes of Operation



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

Source: Table 6.1, Stallings 2014

Electronic Codebook (ECB) Mode



(a) Encryption

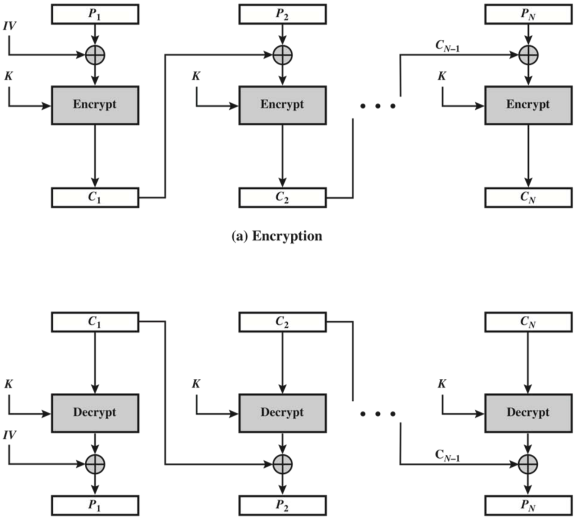
(b) Decryption

Source: Figure 6.3, Stallings 2014

Characteristics of the ECB Mode

- The same 64-bit block of plaintext produces the same ciphertext
  - May subject the encryption algorithm to known plaintext attacks
  - May be vulnerable to modification attacks (substituting or rearranging blocks)
- Ideal only for a short amount of data such as an encryption key

Cipher Block Chaining (CBC) Mode



(a) Encryption

(b) Decryption

Source: Figure 6.4, Stallings 2014

Characteristics of the CBC Mode

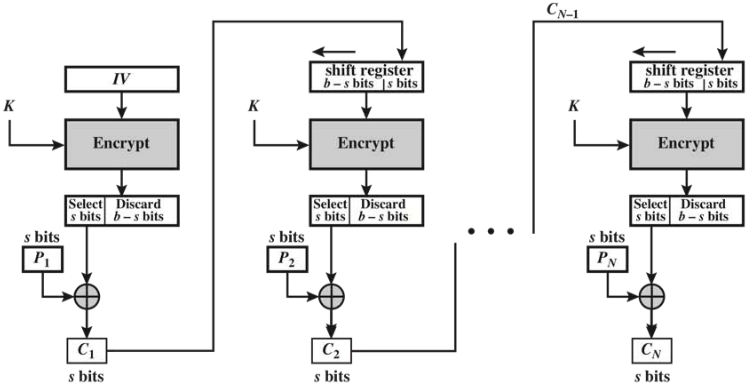
- The Initialization Vector (IV) must be known to both the sender and receiver, and should be protected.
- The opponent may be able to change selected bits of the first block.

$$P_1[i] = IV[i] \oplus D_K(C_1)[i]$$

$$P_1[i]' = IV[i]' \oplus D_K(C_1)[i]$$

- It can also be used for authentication.

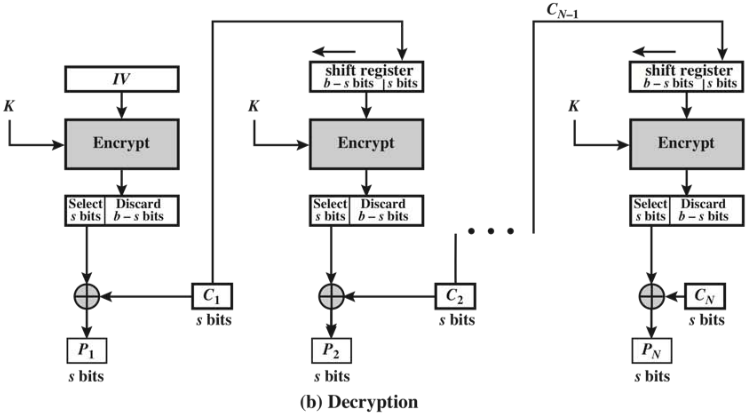
Cipher Feedback (CFB) Mode: Encryption



(a) Encryption

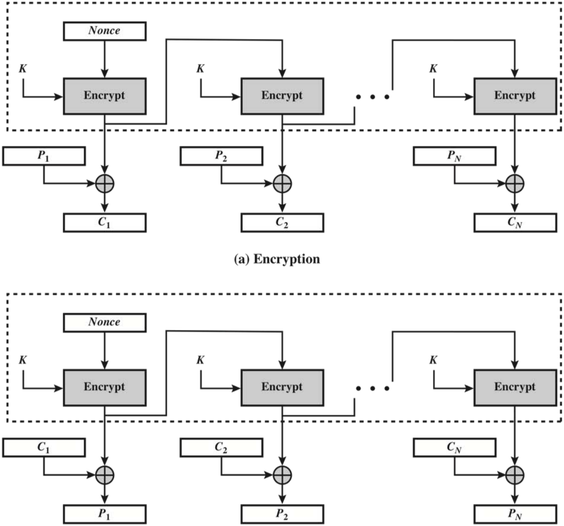
Source: Figure 6.5, Stallings 2014

Cipher Feedback (CFB) Mode: Decryption



Source: Figure 6.5, Stallings 2014

Output Feedback (OFB) Mode



(a) Encryption

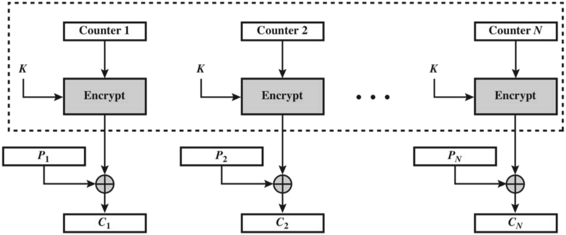
(b) Decryption

Source: Figure 6.6, Stallings 2014

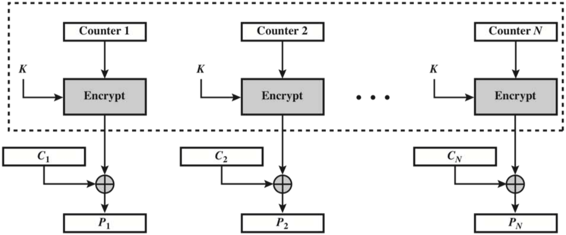
Characteristics of CFB and OFB

- They both can convert a block cipher into a stream cipher.
- Only the encryption function of a cipher is needed.
- In OFB, bit erros in transmission do not propagate.
- OFB is more vulnerable than CFB to a message stream modification attack.
  - For OFB, flipping one bit in the ciphertext will flip the corresponding bit in the recovered plaintext.
  - So, for OFB, controlled changes to the recovered plaintext can be made.

Counter (CTR) Mode



(a) Encryption



(b) Decryption

Source: Figure 6.7, Stallings 2014

Advantages of the CTR MODE

- Hardware/Software efficiency: parallel processing, pipelining, etc.
- Preprocessing: outputs of the encryption boxes
- Random access
- Provable security: as secure as other modes
- Simplicity: similar to CFB and OFB, only the encryption function is needed

Advantages of the CTR MODE

- Hardware/Software efficiency: parallel processing, pipelining, etc.
- Preprocessing: outputs of the encryption boxes
- Random access
- Provable security: as secure as other modes
- Simplicity: similar to CFB and OFB, only the encryption function is needed

Any disadvantage?

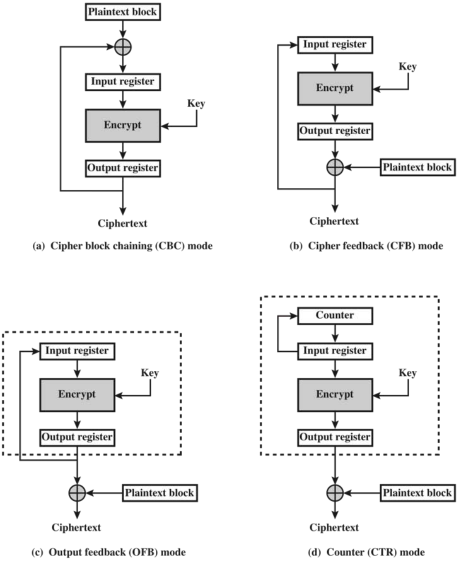
Advantages of the CTR MODE

- Hardware/Software efficiency: parallel processing, pipelining, etc.
- Preprocessing: outputs of the encryption boxes
- Random access
- Provable security: as secure as other modes
- Simplicity: similar to CFB and OFB, only the encryption function is needed

Any disadvantage?

An adversary can flip any bit of the plaintext simply by flipping the corresponding bit of the ciphertext.

Feedback Characteristics

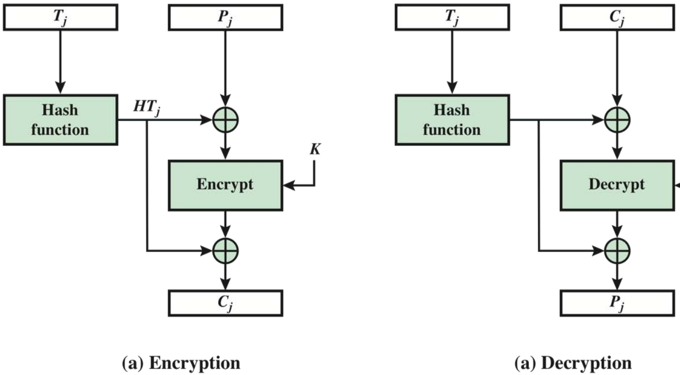


Source: Figure 6.8, Stallings 2014

XTS-AES Mode

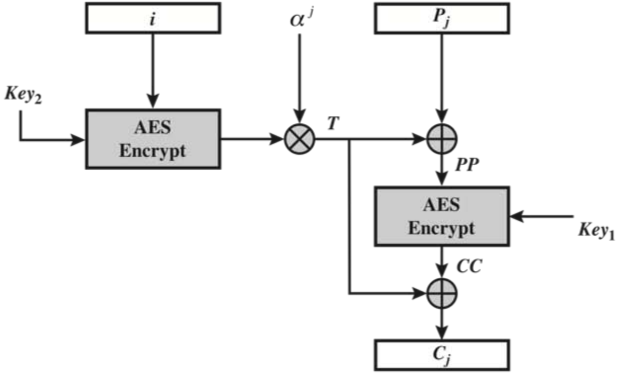
- Approved by NIST in 2010
- Also an IEEE standard, IEEE Std 1619-2007
- For data stored in sector-based devices
- Based on the concept of a tweakable block cipher
- Plaintext organized into blocks of 128 bits
- Like the ECB mode but a different tweak value used for each block
- “ciphertext-stealing” used instead of padding (when the last block is less than 128 bits long)

Tweakable Block Ciphers



Source: Figure 6.9, Stallings 2014

XTS-AES Operation on Single Block: Encryption

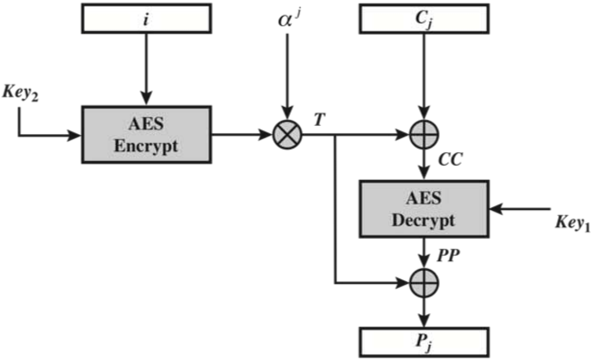


(a) Encryption

Note: $\otimes$ is multiplication in $GF(2^{128})$ with $x^{128} + x^7 + x^2 + x + 1$ as the irreducible modulus.

Source: Figure 6.10, Stallings 2014

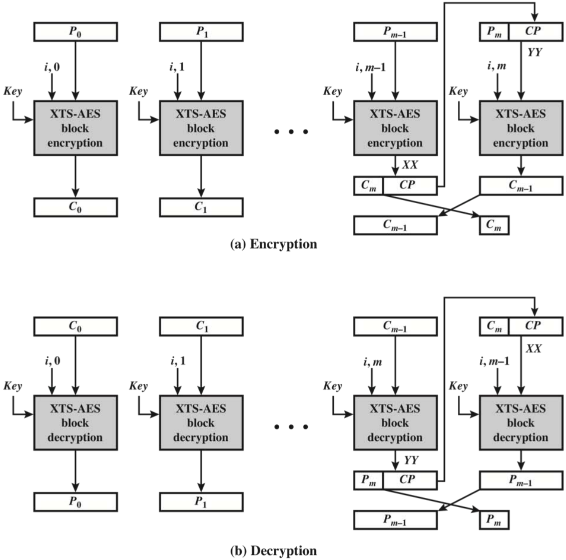
XTS-AES Operation on Single Block: Decryption



(b) Decryption

Source: Figure 6.10, Stallings 2014

XTS-AES Operation on a Sector



(a) Encryption

(b) Decryption

Source: Figure 6.11, Stallings 2014