Modes of Operation

Sachin Tripathi

IIT(ISM), Dhanbad

Outline

Overview of Modes of Operation
Security Pitfalls/advantage of each Mode of Operation

Introduction

Typical Block ciphers encrypt fixed size blocks like DES encrypts 64-bit blocks, and AES encrypts 128-bit blocks.
In practice, typical volume of the message is more than conventional block size.

- Divide the message into number of blocks for ciphering one by one.
- A mode of operation describes the process of enciphering each of the block under a single key.

Modes of Operation

Current well-known modes of operation

- Electronic Code Book mode (ECB),
- Cipher Block Chaining mode (CBC),
- Cipher Feedback mode (CFB),
- Output Feedback mode (OFB),
- Counter mode (CTR),
- Galois Counter mode (GCM)

Terminology

Initialize Vector (IV)

- a block of bits to randomize the encryption and hence to produce distinct ciphertext

Nonce : Number (used) Once

- Random of psuedorandom number to ensure that past communications can not be reused in replay attacks
- Some also refer to initialize vector as nonce

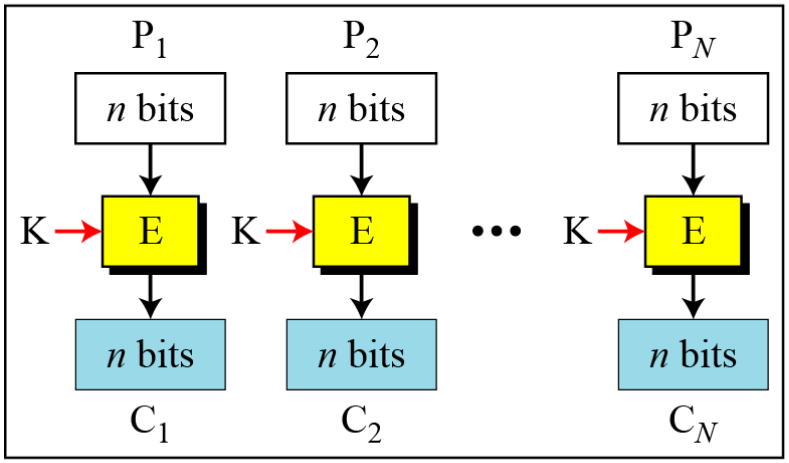
Padding

- final block may require a padding to fit a block size

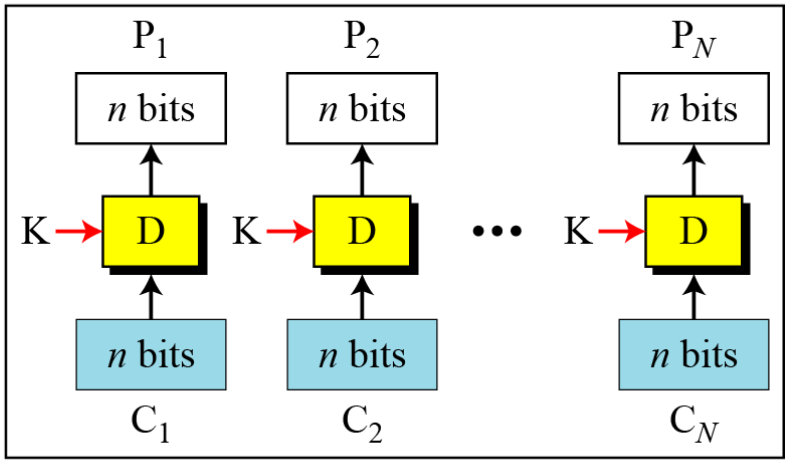
Electronic Codebook Mode (ECB)

Encryption: $C_i = E_K(P_i)$ Decryption: $P_i = D_K(C_i)$

E: Encryption D: Decryption
P_i: Plaintext block i C_i: Ciphertext block i
K: Secret key



Encryption



Decryption

Ciphertext Stealing

In ECB mode, padding must be added to the last block if it is not n bits long. Padding is not always possible. For example, when the ciphertext needs to be stored in the buffer where the plaintext was previously stored, plaintext and ciphertext must be the same. A technique called **ciphertext stealing (CTS)** can make it possible to use ECB mode without padding. In this technique the last two plaintext blocks, P_{N-1} and P_N, are encrypted differently and out of order, as shown below, assuming that P_{N-1} has n bits and P_N has m bits, where $m \leq n$.

$$X = E_K(P_{N-1}) \quad \rightarrow \quad C_N = head_m(X)$$
$$Y = P_N \mid tail_{n-m}(X) \quad \rightarrow \quad C_{N-1} = E_K(Y)$$

The $head_m$ function selects the leftmost m bits; the $tail_{n-m}$ function selects the rightmost $n - m$ bits. The detailed diagram and the procedure of the encryption and decryption are left as exercises.

Security Issues

Following are security issues in CBC mode:

1. Patterns at the block level are preserved. For example, equal blocks in the plaintext become equal blocks in the ciphertext. If Eve finds out that ciphertext blocks 1, 5,and 10 are the same, she knows that plaintext blocks 1, 5, and 10 are the same. This is a leak in security. For example, Eve can do an exhaustive search to decrypt only one of these blocks to find the contents of all of them.

2. The block independency creates opportunities for Eve to exchange some ciphertext blocks without knowing the key. For example, if she knows that block 8 always conveys some specific information, she can replace this block with the corresponding block in the previously intercepted message.

Example

Assume that Eve works in a company a few hours per month (her monthly payment is very low). She knows that the company uses several blocks of information for each employee in which the seventh block is the amount of money to be deposited in the employee's account. Eve can intercept the ciphertext sent to the bank at the end of the month, replace the block with the information about her payment with a copy of the block with the information about the payment of a full-time colleague. Each month Eve can receive more money than she deserves.

Error Propagation

A single bit error in transmission can create errors in several (normally half of the bits or all of the bits) in the corresponding block. However, the error does not have any effect on the other blocks.

Advantages of ECB mode

- Block synchronization between the encryption and decryption parties Alice and Bob is not necessary.
- Bit errors, e.g., caused by noisy transmission lines, only affect the corresponding block but not succeeding blocks.
- ECB mode can be parallelized.

Limitations

- It encrypts highly deterministically. This means that identical plaintext blocks result in identical ciphertext blocks, as long as the key does not change.
- The ECB mode is susceptible to *substitution attacks*.

Block #	1	2	3	4	5
	Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount $

Example for a substitution attack against ECB encryption

Applications

- The ECB mode of operation is not recommended for encryption of messages of more than one block to be transferred through an insecure channel. If the message is short enough to fit in one block, the security issues and propagation errors are tolerable.

- One area where the independency of the ciphertext block is useful is where records need to be encrypted before they are stored in a database or decrypted before they are retrieved.

- Because the order of encryption and decryption of blocks is not important in this mode, access to the database can be random if each record is a block or multiple blocks. A record can be retrieved from the middle, decrypted, and encrypted after modification without affecting other records.

CRYPTOGRAPHY AND DATA SECURITY

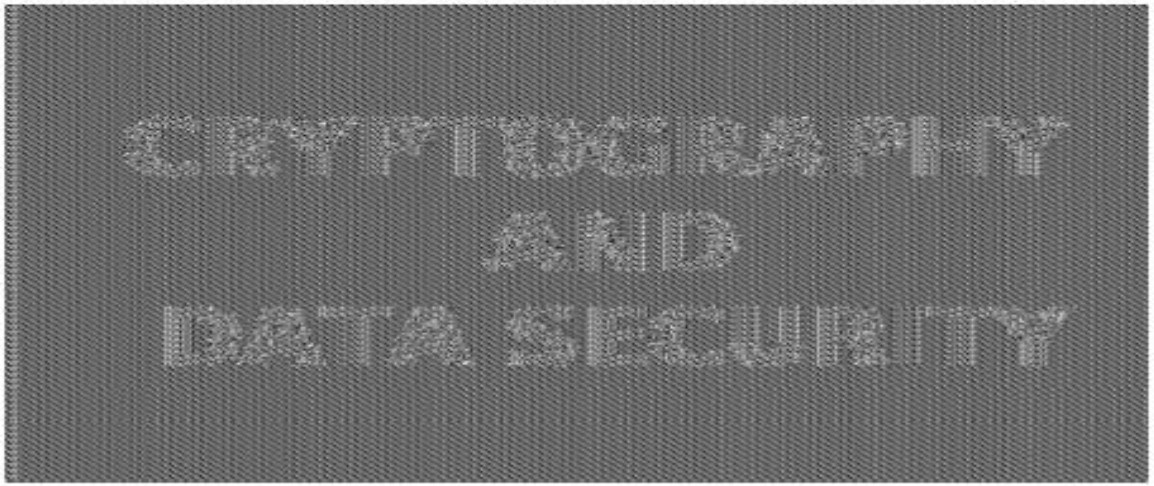
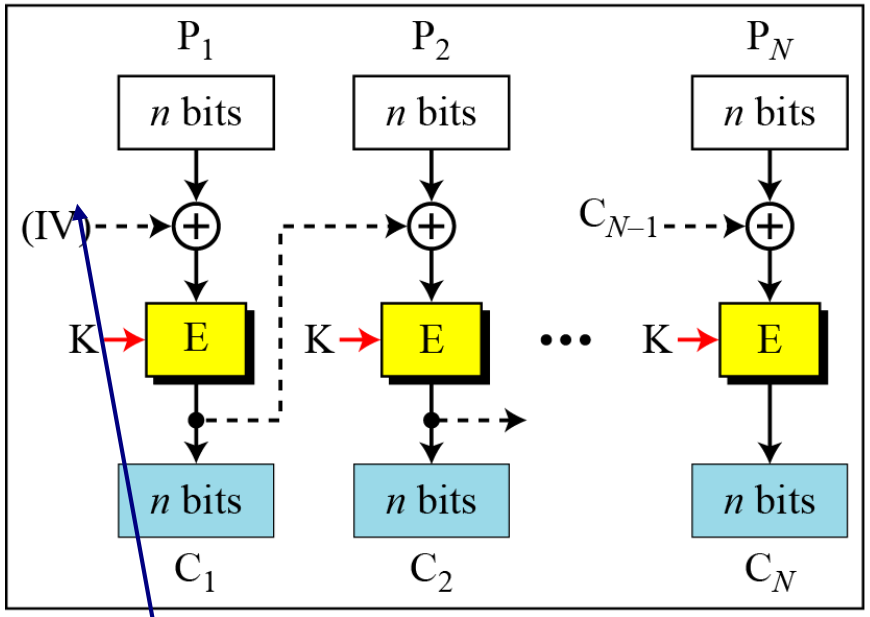


Image and encrypted image using AES with 256-bit key in ECB mode

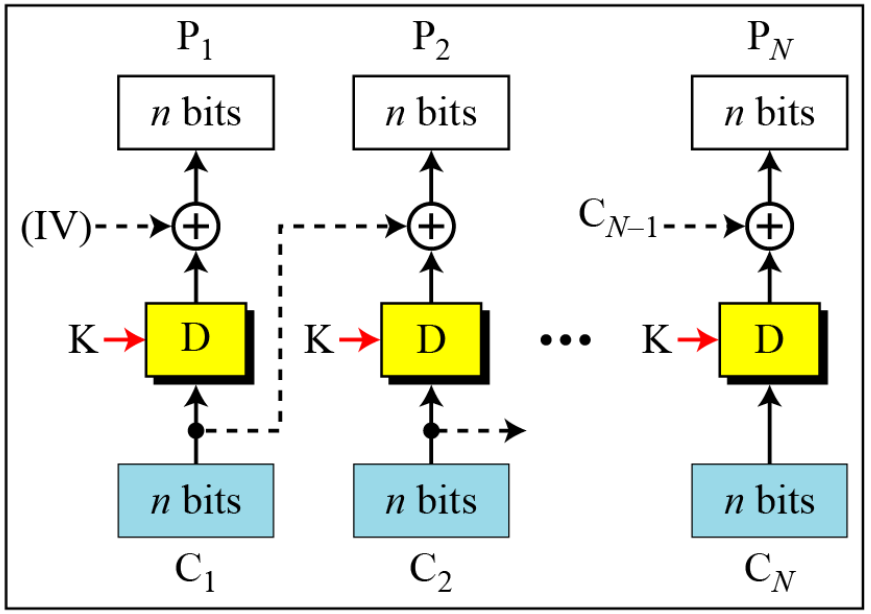
Cipher Block Chaining (CBC)

E: Encryption
P_i: Plaintext block i
K: Secret key

D : Decryption
C_i : Ciphertext block i
IV: Initial vector (C_0)



Encryption



Decryption

Encryption:
$C_0 = IV$
$C_i = E_K (P_i \oplus C_{i-1})$

Decryption:
$C_0 = IV$
$P_i = D_K (C_i) \oplus C_{i-1}$

$$P_i = D_K (C_i) \oplus C_{i-1} = D_K (E_K (P_i \oplus C_{i-1})) \oplus C_{i-1} = P_i \oplus C_{i-1} \oplus C_{i-1} = P_i$$

Discussion

- The encryption of a block depends on the current and **all** blocks before it. So, repeated plaintext blocks are encrypted differently.
- Initialization Vector (IV)
  - Must be known to both the sender & receiver
  - Typically, IV is either a fixed value or is sent encrypted in ECB mode before the rest of ciphertext.
- Decryption is message dependent.
- Error propagation does occur.

CBC Encryption

```
CBC_Encryption (IV, K, Plaintext blocks)
{
    C_0 ← IV
    for (i = 1 to N)
    {
        Temp ← P_i ⊕ C_{i-1}
        C_i ← E_K (Temp)
    }
    return Ciphertext blocks
}
```

Security Issues

Following are two of the security issues in CBC mode:

- In CBC mode, equal plaintext blocks belonging to the same message are enciphered into different ciphertext blocks. In other words, the patterns at the block levels are not preserved.
  - However, if two messages are equal, their Encipherment is the same if they use the same IV. As a matter of fact, if the first M blocks in two different messages are equal, they are enciphered into equal blocks unless different IVs are used. For this reason, some people recommend the use of a timestamp as an IV.

- Eve can add some ciphertext blocks to the end of the ciphertext stream.

Error Propagation

In CBC mode, a single bit error in ciphertext block C_j during transmission may create error in most bits in plaintext block P_j during decryption. However, this single error toggles only one bit in plaintext block P_{j+1} (the bit in the same location). The proof of this fact is left as an exercise. Plaintext blocks P_{j+2} to P_N are not affected by this single bit error. A single bit error in ciphertext is *self-recovered*.

Ciphertext Stealing

The ciphertext stealing technique described for ECB mode can also be applied to CBC mode, as shown below

$$\begin{array}{lllll} U = P_{N-1} \oplus C_{N-2} & \rightarrow & X = E_K(U) & \rightarrow & C_N = head_m(X). \\ V = P_N \mid pad_{n-m}(0) & \rightarrow & Y = X \oplus V & \rightarrow & C_{N-1} = E_K(Y) \end{array}$$

The *head* function is the same as described in ECB mode; the *pad* function inserts 0's.

Applications

The CBC mode of operation can be used to encipher messages. However, because of chaining mechanism, parallel processing is not possible.

CBC mode is not used to encrypt and decrypt random-access Þles records because encryption and decryption require access to the previous records. As we will see in Chapter 11, CBC mode is also
used for authentication.

Cipher FeedBack

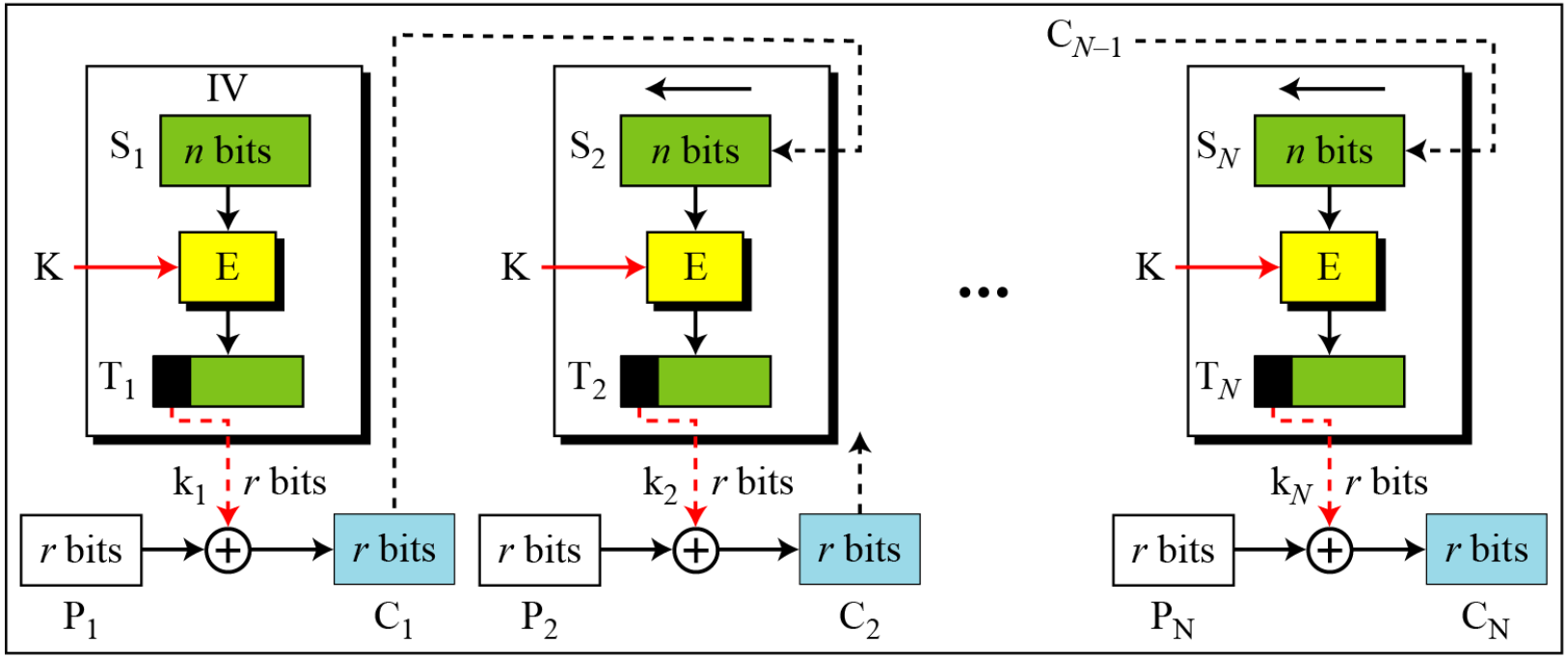
Encryption: $C_i = P_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$

Decryption: $P_i = C_i \oplus \text{SelectLeft}_r \{E_K [\text{ShiftLeft}_r (S_{i-1}) \mid C_{i-1})]\}$

E : Encryption
P_i: Plaintext block i
K: Secret key

D : Decryption
C_i: Ciphertext block i
IV: Initial vector (S_1)

S_i: Shift register
T_i: Temporary register



Encryption

CFB as Stream Cipher

- Although CFB is an operation mode for using block ciphers such as DES or AES, the result is a stream cipher. In fact, it is a nonsynchronous stream cipher in which the key stream is dependent on the ciphertext.

CFB Encryption

```
CFB_Encryption (IV, K, r)
{
   i ← 1
   while (more blocks to encrypt)
   {
   input (P_i)
   if (i = 1)
      S ← IV
   else
      {
      Temp ← shiftLeft_r (S)
      S ← concatenate (Temp, C_{i-1})
      }
   T ← E_K(S)
   k_i ← selectLeft_r (T)
   C_i ← P_i ⊕ k_i
   output (C_i)
   i ← i + 1
   }
}
```

Security Issues

There are three primary security issues in CFB mode:

- Just like CBC, the patterns at the block level are not preserved.
- More than one message can be encrypted with the same key, but the value of the IV should be changed for each message. This means that Alice needs to use a different IV each time she sends a message.
- Eve can add some ciphertext block to the end of the ciphertext stream.

Error Propagation

- In CFB, a single bit error in ciphertext block Cj during transmission creates a single bit error (at the same position) in plaintext block Pj.

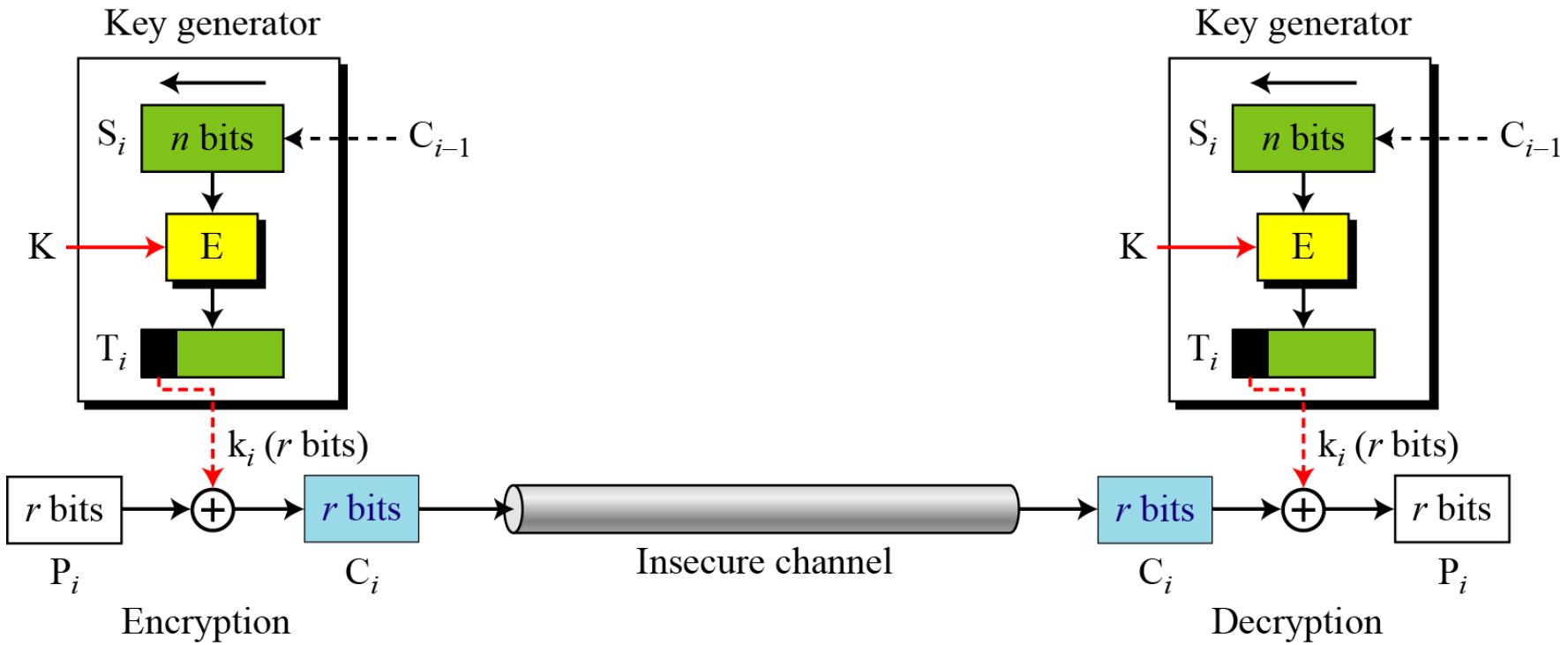
- However, most of the bits in the following plaintext blocks are in error (with 50 percent probability) as long as some bits of Cj are still in the shift register. The calculation of the number of affected blocks is left as an exercise. After the shift register is totally refreshed, the system recovers from the error.

Applications

The CFB mode of operation can be used to encipher blocks of small size such as one character or bit at a time. There is no need for padding because the size of the plaintext block is normally fixed (8 for a character or 1 for a bit).

CFB as a Stream Cipher

In CFB mode, encipherment and decipherment use the encryption function of the underlying block cipher.



Discussion on CFB

A very strange feature of CFB mode is that we actually decrypt the ciphertext using only the encryption process of the block cipher.

The encryption algorithm is never used directly to encrypt the plaintext, but is rather used as keystream generator to produce keystream that is placed in the bottom register.

It gives the direction to use the CFB mode as a stream cipher.

Advantage

Same block will not produce the same cipher text.
Padding is not essential. (Suppose 400 bits plaintext is encrypted using a block cipher with a 128 bits block length. Last block having 16 bits content. It is possible to encipher last 16 bits here.)

Limitation

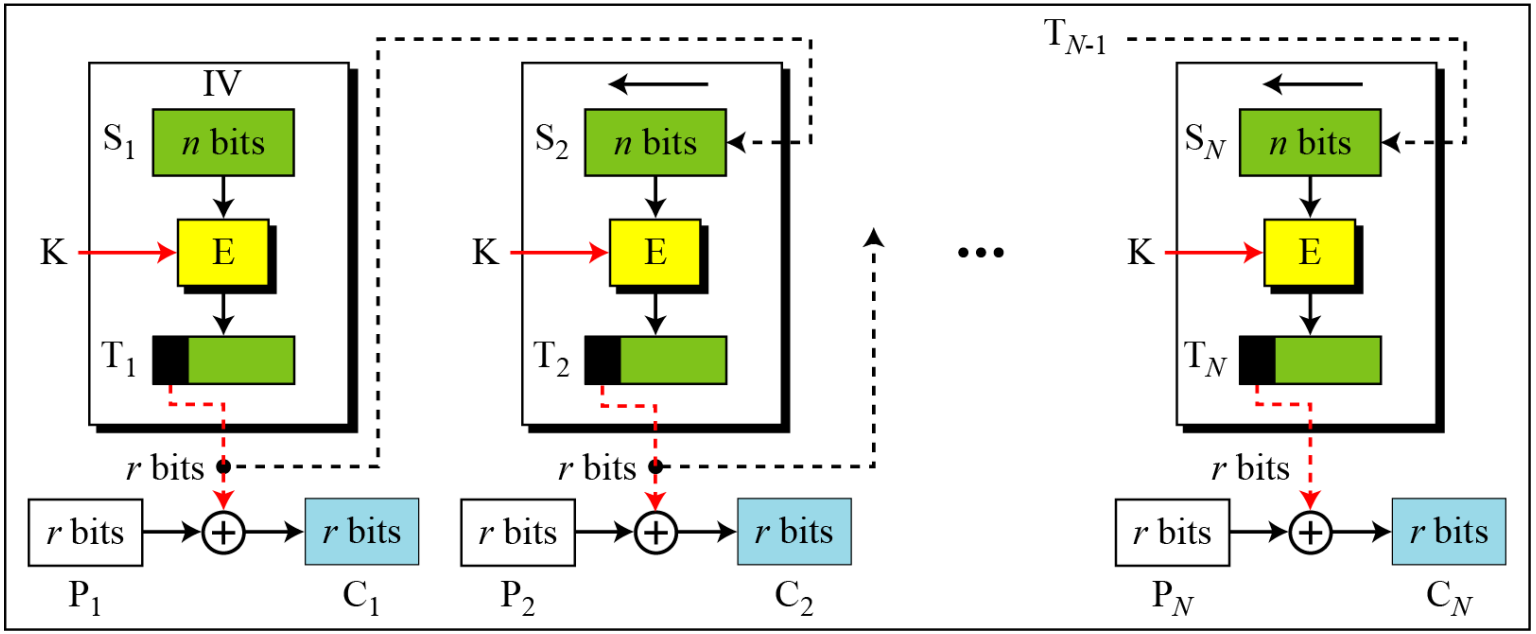
- Decryption process depend on the entire preceding encryption process .
- Error propagation does occur.

Output FeedBack

E : Encryption
P_i: Plaintext block i
K: Secret key

D : Decryption
C_i: Ciphertext block i
IV: Initial vector (S_1)

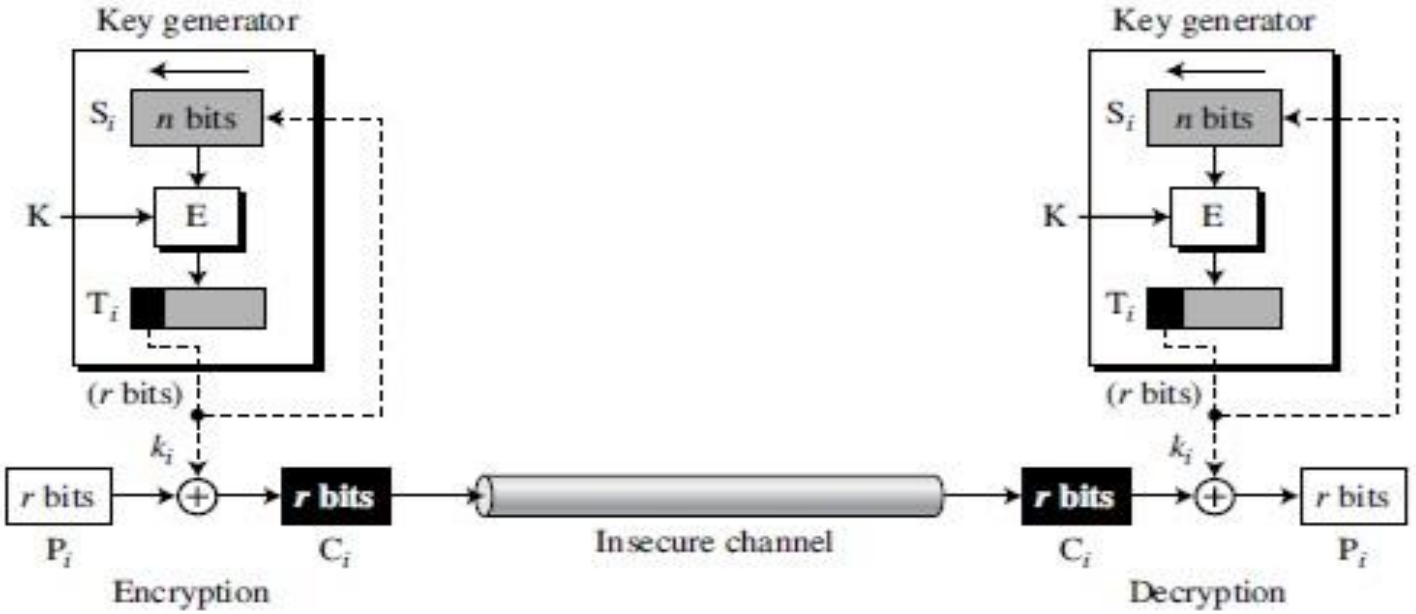
S_i: Shift register
T_i: Temporary register



Encryption

It avoids error propagation.

OFB as a stream cipher



Encryption

```
OFB_Encryption (IV, K, r)
{
    i ← 1
    while (more blocks to encrypt)
    {
        input (P_i)
        if (i = 1)   S ← IV
        else
        {
            Temp ← shiftLeft_r (S)
            S ← concatenate (Temp, k_{i-1})
        }
        T ← E_K (S)
        k_i ← selectLeft_r (T)
        C_i ← P_i ⊕ k_i
        output (C_i)
        i ← i + 1
    }
}
```

OFB Scheme

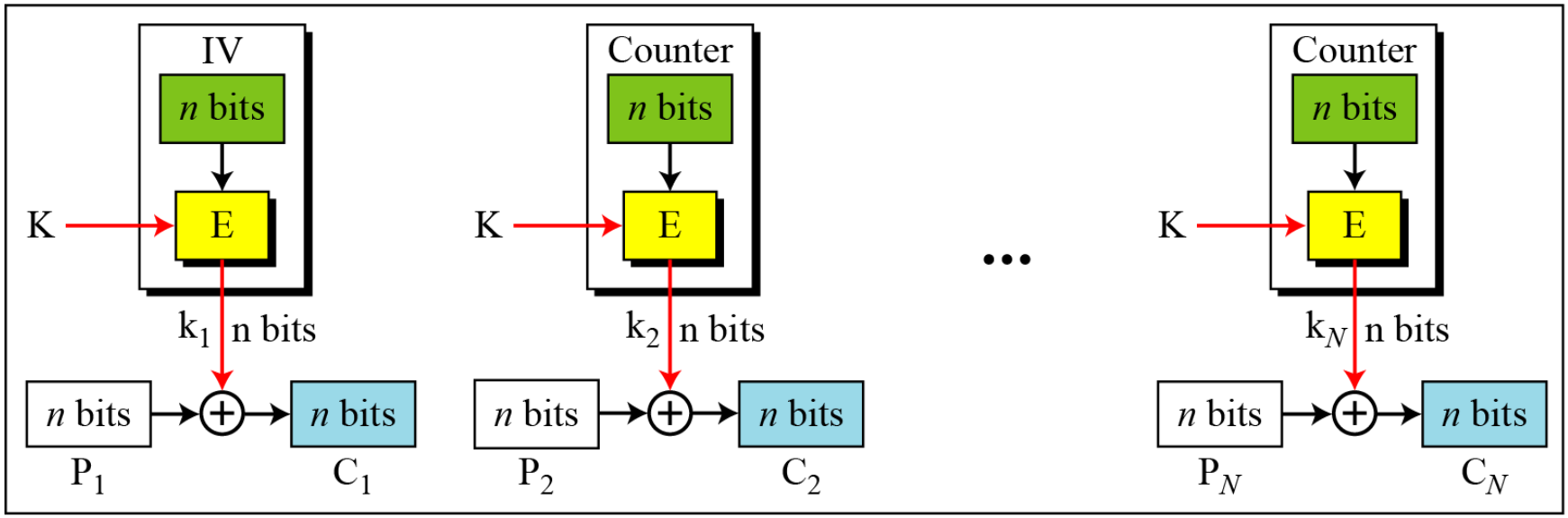
- Output feedback (OFB) mode is very similar to CFB mode, with one difference: each bit in the ciphertext is independent of the previous bit or bits. This avoids error propagation. If an error occurs in transmission, it does not affect the bits that follow.

CTR Scheme

E : Encryption
P_i : Plaintext block *i*
K : Secret key

IV: Initialization vector
C_i : Ciphertext block *i*
k_i : Encryption key *i*

The counter is incremented for each block.



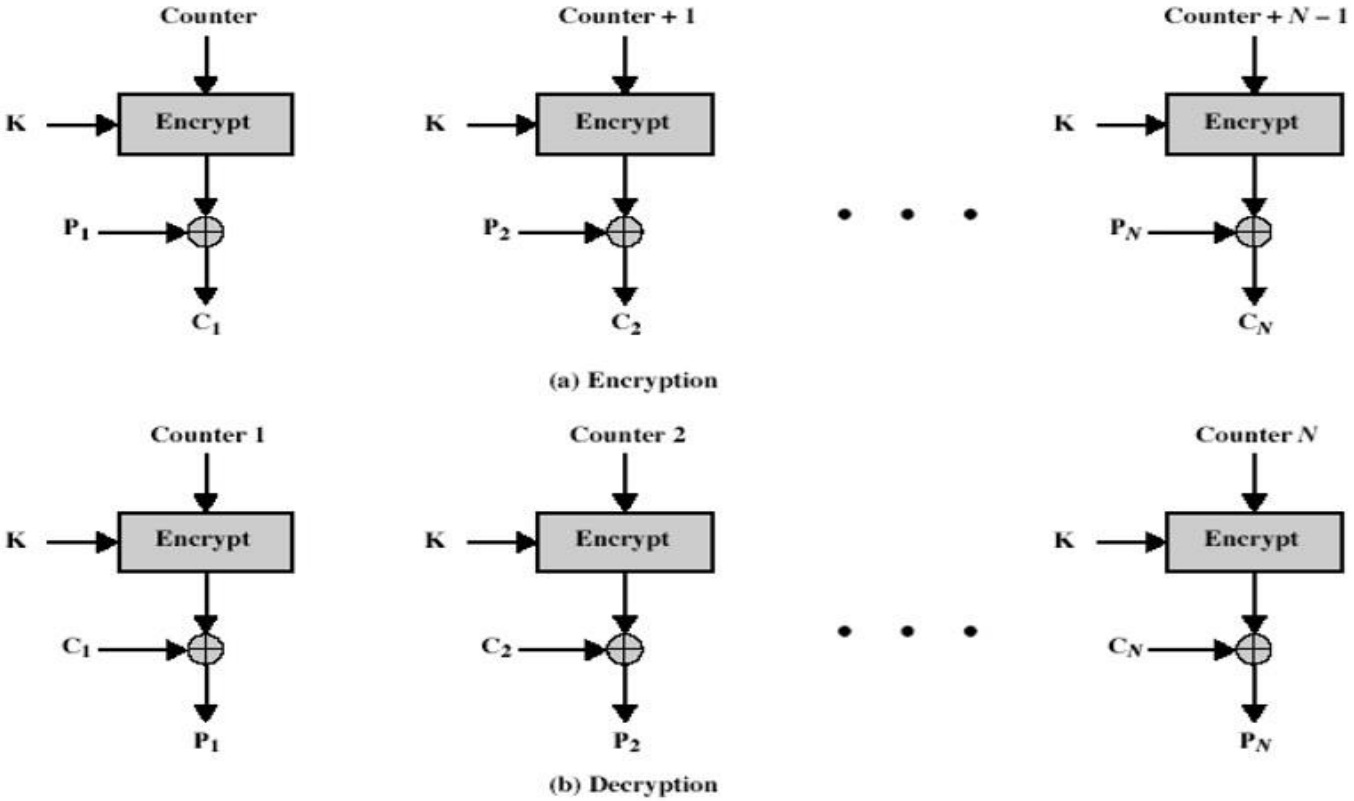
Encryption

Discussion

CTR mode does not have message dependency.
It can be parallelized.

- Needs only the encryption algorithm

CTR Encryption and Decryption



CBC and CTR comparison

CBC	CTR
Padding needed	No padding
No parallel processing	Parallel processing
Separate encryption and decryption functions	Encryption function alone is enough
Random IV or a nonce	Unique nonce
Nonce reuse leaks some information about initial plaintext block	Nonce reuse will leak information about the entire message

Comparison of Different Modes

Operation Mode	*Description*	*Type of Result*	*Data Unit Size*
ECB	Each n-bit block is encrypted independently with the same cipher key.	Block cipher	n
CBC	Same as ECB, but each block is first exclusive-ored with the previous ciphertext.	Block cipher	n
CFB	Each r-bit block is exclusive-ored with an r-bit key, which is part of previous cipher text	Stream cipher	$r \leq n$
OFB	Same as CFB, but the shift register is updated by the previous r-bit key.	Stream cipher	$r \leq n$
CTR	Same as OFB, but a counter is used instead of a shift register.	Stream cipher	n

Galois Counter mode (GCM)

- All of the five modes have one goal: They encrypt data and thus provide confidentiality for a message sent from Alice to Bob.

- In practice, we often not only want to keep data confidential, but Bob also wants to know whether the message is really coming from Alice.
- This is called authentication and the Galois Counter mode (GCM) is a mode of operation that lets the receiver (Bob) determine whether the message was really sent by the person he shares a key with (Alice).

Contd…

- The *Galois Counter Mode (GCM) is an encryption mode which also computes a* message authentication code (MAC).

- MAC provides a cryptographic checksum that is computed by the sender, Alice, and appended to the message. Bob also computes a MAC from the message and checks whether his MAC is the same as the one computed by Alice.
- This way, Bob can make sure that (1) the message was really created by Alice and (2) that nobody tampered with the ciphertext during transmission.

Contd...

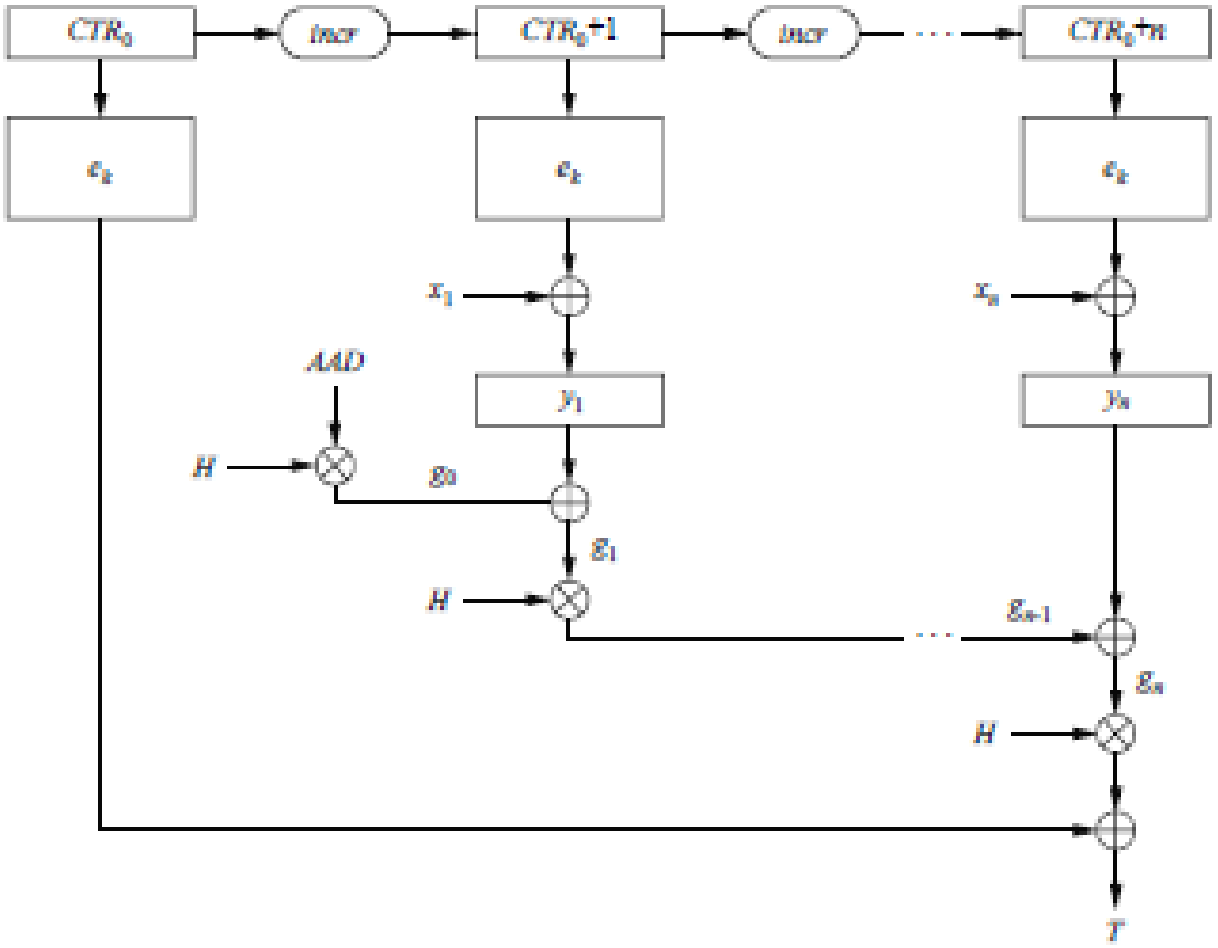
For authentication, GCM performs a chained Galois field multiplication.

All multiplications are in the 128-bit Galois field $GF(2^{128})$ with the irreducible polynomial $P(x) = x^{128}+x^7+x^2+x+1$.

GCM encrypts data using the Counter Mode (CTR) followed by the computation of a MAC value.

Initial authenticated data is considered and that is known as AAD called additional authenticated data.

Contd...



Basic authenticated encryption in Galois Counter mode

Contd...

Definition Basic Galois Counter mode (GCM)
Let $e()$ be a block cipher of block size 128 bit; let x be the plaintext consisting of the blocks $x_1, \ldots, x_n$; and let AAD be the additional authenticated data.

1. ***Encryption***

   a. *Derive a counter value CTR_0 from the IV and compute $CTR_1 = CTR_0 + 1$.*
   b. *Compute ciphertext: $y_i = e_k(CTR_i) \oplus x_i, \quad i \geq 1$*

2. ***Authentication***

   a. *Generate authentication subkey $H = e_k(0)$*
   b. *Compute $g_0 = AAD \times H$ (Galois field multiplication)*
   c. *Compute $g_i = (g_{i-1} \oplus y_i) \times H, \quad 1 \leq i \leq n$ (Galois field multiplication)*
   d. *Final authentication tag: $T = (g_n \times H) \oplus e_k(CTR_0)$*

Contd...

- The receiver of the packet $[(y_1, \ldots, y_n), T, AAD]$ decrypts the ciphertext by applying the Counter mode.
- To check the authenticity of the data, the receiver also computes an authentication tag T' using the received ciphertext and *AAD* as input.
- Receiver employs exactly the same steps as the sender.

If T' and T match, the receiver is assured that the ciphertext (and *ADD*) were not manipulated in transit and that only the sender could have generated the message.

Conclusions

- There are many different ways to encrypt with a block cipher. Each mode of operation has some advantages and disadvantages.

Several modes turn a block cipher into a stream cipher.

- The straightforward ECB mode has security weaknesses, independent of the underlying block cipher.
- The counter mode allows parallelization of encryption and is thus suited for high speed implementations.
- Double encryption with a given block cipher only marginally improves the resistance against brute-force attacks.