Cryptography II

Block ciphers and modes of operations

Block ciphers: getting the concept

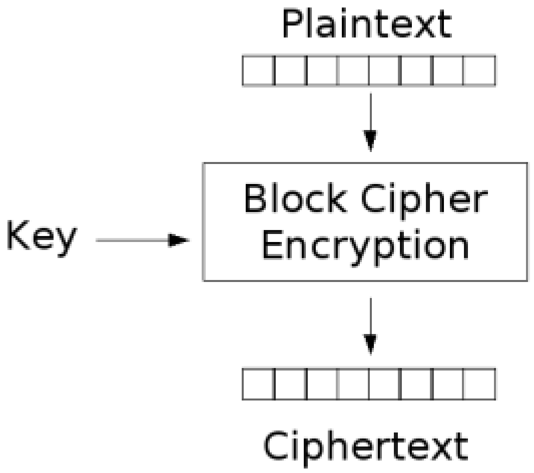
- Stream cipher vs. block cipher: single unit vs. block of units

key	000	001	010	011	100	101	110	111
0	001	111	110	000	100	010	101	011
1	001	110	111	100	011	010	000	101
2	001	000	100	101	110	111	010	011
3	100	101	110	111	000	001	010	011
4	101	110	100	010	011	001	011	111

- Plaintext= 010100110111= (010)(100)(110)(111)
  - ➔ Ciphertext = 111 011 000 101 theo key=1
  - ➔ Ciphertext = 100 011 011 111 theo key=4
- There are 5 keys, $2^2 < 5 < 2^3$ ➔ need keys in 3 bits to present➔ key size= block size= 3.
- Small sizes are dangerous, however: If Eve catches C=001 ➔ can infer P= 000 or 101.

Block cipher: an invertible map

- Map n-bit plaintext blocks to n-bit ciphertext blocks (n: block size/length).
- For n-bit plaintext and ciphertext blocks and a fixed key, the encryption function is a bijection:
  - $E : P_n \times K \rightarrow C_n$ s.t. for all key $k \in K$, $E(x, k)$ is an invertible mapping written $E_k(x)$.
- The inverse mapping is the decryption function, $y = D_k(x)$ denotes the decryption of plaintext x under k.



General condition in creating secure block ciphers

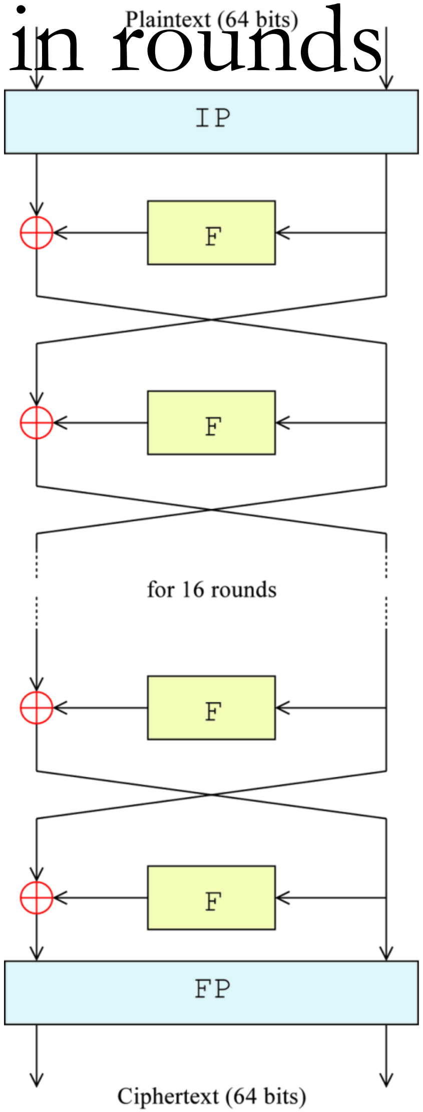
- The block size has to be large enough to prevent against statistical analysis
  - However, larger block size means slower processing
- The key space (then key length) must be large enough to prevent against exhaustive key search
  - However, key length shouldn’t be too big that makes key distribution and management more difficult

General principles in designing secure block ciphers

- *Confusion:* As a function, the dependence of the ciphertext on the plaintext should be complex enough so that enemy can’t find the rules
  - The function should be non-linear.
- *Diffusion:* The goal is to spread the information from the plaintext over the entire ciphertext so that changes in plaintext affect many parts in ciphertext
  - This makes it difficult for an enemy to break the cipher by using statistical analysis
- Confusion is made by usings substitutions while *diffusion* by transpositions and/or permutations.

The Feistel structure: processing in rounds

- Block ciphers are usually designed with many rounds where basic round accomplishes the core function f for basic confusion and diffusion.
  - The input of a round is the output of the previous round and a subkey which is generated by a key-schedule algorithm
- The decryption is a reverse process where the sub-keys are handled in the reverse order



The overall Feistel structure of DES

Block Ciphers Features

- Block size**:** in general larger block sizes mean greater security.
- Key size**:** larger key size means greater security (larger key space).
- Number of rounds**:** multiple rounds offer increasing security.
- Encryption modes: define how messages larger than the block size are encrypted, very important for the security of the encrypted message.

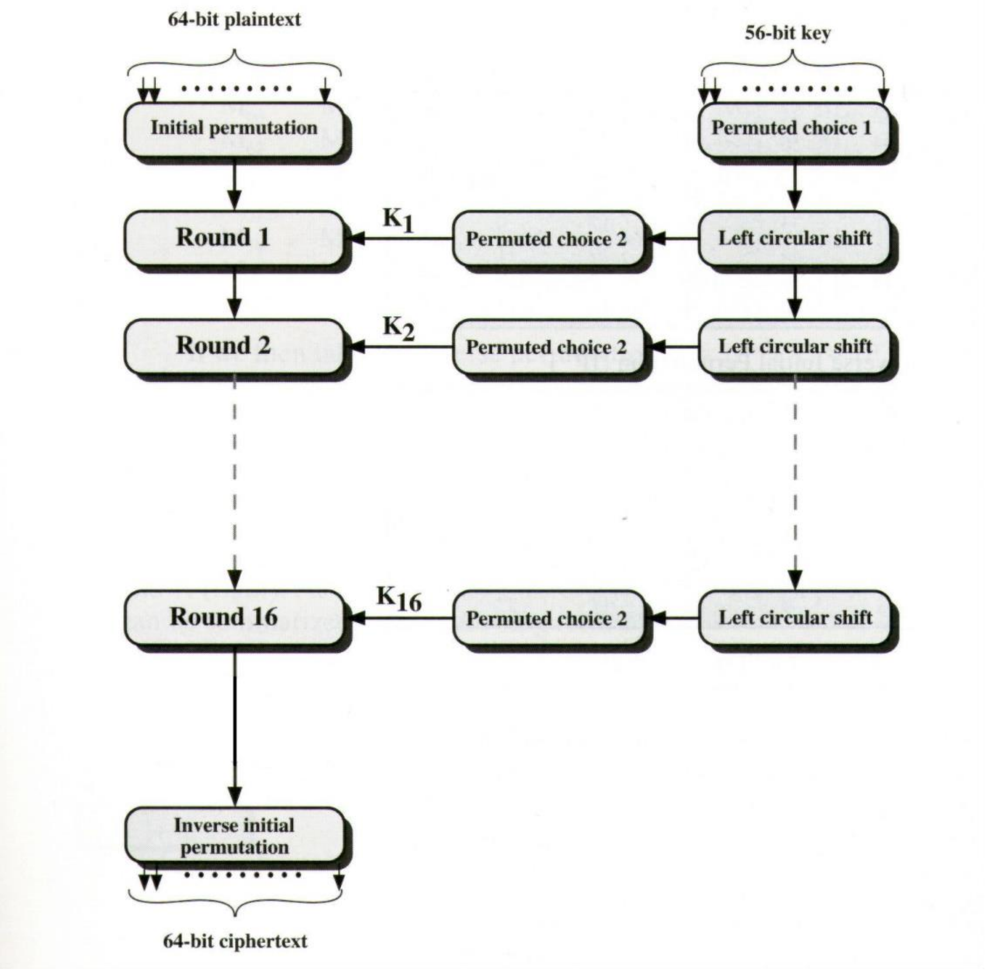
History of Data Encryption Standard (DES)

- 1967: Feistel at IBM
  - Lucifer: block size 128; key size 128 bit
- 1972: NBS asks for an encryption standard
- 1975: IBM developed DES (modification of Lucifer
  - block size 64 bits; key size 56 bits
- 1975: NSA suggests modification
- 1977: NBS adopts DES as encryption standard in (FIPS 46-1, 46-2).
- 2001: NIST adopts Rijndael as replacement to DES

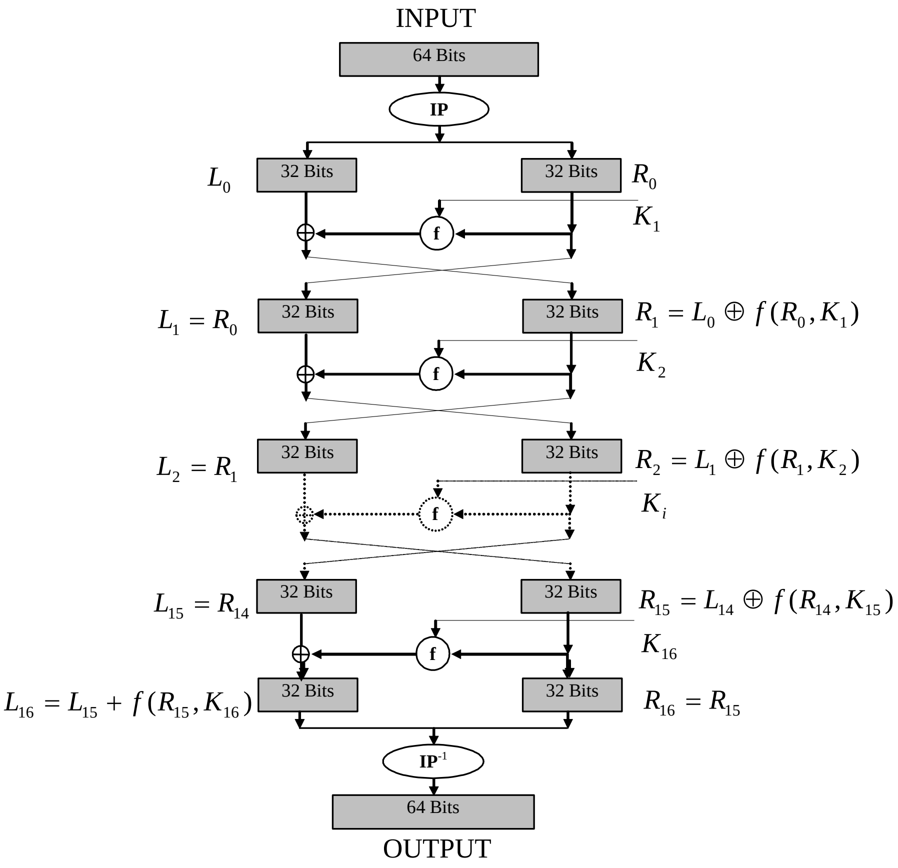
DES Features

- **Features:**
  - **Block size = 64 bits**
  - **Key size = 56 bits**
  - **Number of rounds = 16**
  - **16 intermediary keys, each 48 bits**

DES Rounds


64-bit plaintext
56-bit key
Initial permutation
Permuted choice 1
Round 1
K1
Permuted choice 2
Left circular shift
Round 2
K2
Permuted choice 2
Left circular shift
Round 16
K16
Permuted choice 2
Left circular shift
Inverse initial permutation
64-bit ciphertext

DES encryption: A closer look



Decryption uses the same algorithm as encryption, except that the subkeys K_1, K2, …K16 are applied in reversed order

Cryptanalysis of DES

Brute Force:

- Known-Plaintext Attack
- Try all 2^{56} possible keys
- Requires constant memory
- Time-consuming
- DES challenges: (RSA)
  - msg="the unknown message is :xxxxxxxx"
  - CT=" C1 | C2 | C3 | C4"
  - 1997 Internet search: 3 months
  - 1998 EFF machine (costs $250K): 3 days
  - 1999 Combined: 22 hours

Rijndael Features

- Designed to be efficient in both hardware and
- software across a variety of platforms.
- Uses a variable block size, **128,192, 256-bits**, key size **of 128-, 192-, or 256-bits.**
- 128-bit round key used for each round (Can be precomputed and cached for future encryptions).
- Note: AES uses a 128-bit block size.
- Variable number of rounds (10, 12, 14):
  - 10 if B = K = 128 bits
  - 12 if either B or K is 192 and the other is ≤ 192
  - 14 if either B or K is 256 bits

Rijndael Design

- Operations performed on State (4 rows of bytes).
- The 128 bit key is expanded as an array of 44 32bits words; 4 distinct words serve as a round key for each round; key schedule relies on the S-box
- Algorithms composed of three layers
  - Linear diffusion
  - Non-linear diffusion
  - Key mixing

Decryption

- The decryption algorithm is not identical with the encryption algorithm, but uses the same key schedule.
- There is also a way of implementing the decryption with an algorithm that is equivalent to the encryption algorithm (each operation replaced with its inverse), however in this case, the key schedule must be changed.

Rijandel Cryptanalysis

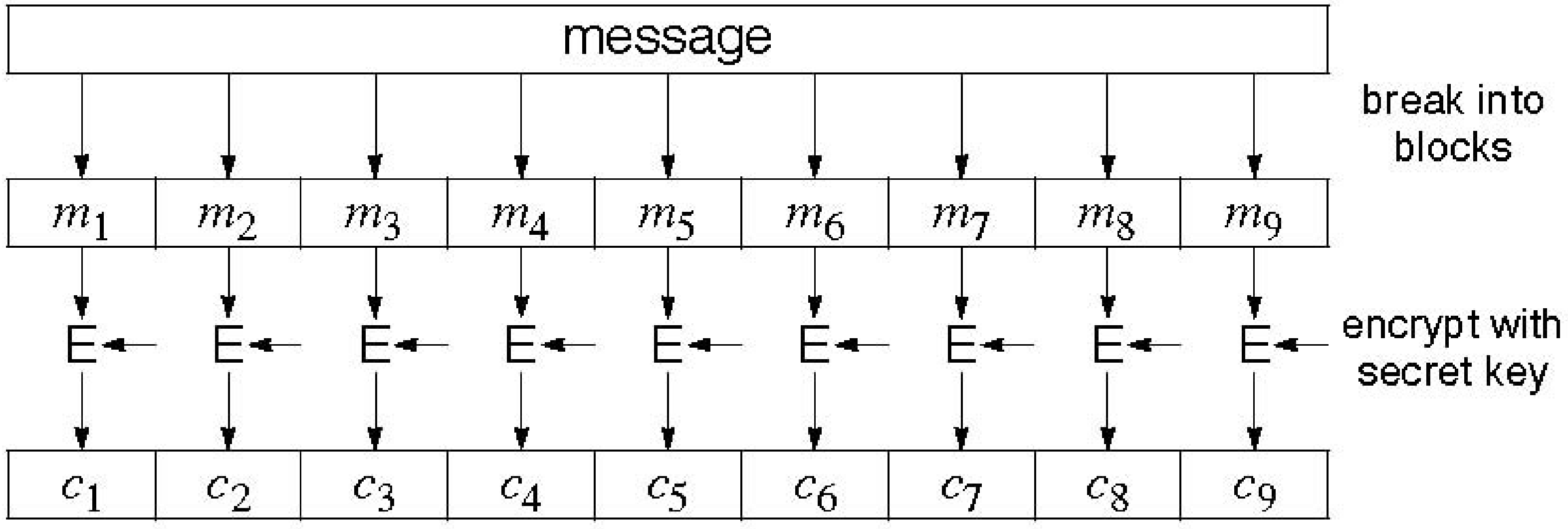
- Academic break on weaker version of the cipher, 9 rounds
- Requires 2^{224} work and 2^{85} chosen *related-key* plaintexts.
- Attack not practical.

Modes of Operation (Encryption modes)

- Mode of operation (or encryption mode):
  - A block cipher algorithm takes on a fixed-length input, i.e. a block, and produce an output, usually a block of the same fixed-length.
  - In practice, we want to encrypt files of various length ➔ need to divide a file into block of that given fixed length ➔ then call the encryption algorithms several times
  - Operation mode: the manner and structure in which we feed the encryption algorithm (several times) with blocks of the plaintext file and concatenate the resulted blocks to produce the ciphertext file.
- The popular modes:
  - ECB, CBC, OFB, CFB, CTR
- We now overview the properties of certain modes (privacy, integrity) and potential attacks against them.

Electronic Code Book (ECB)

- Each block is independently encoded



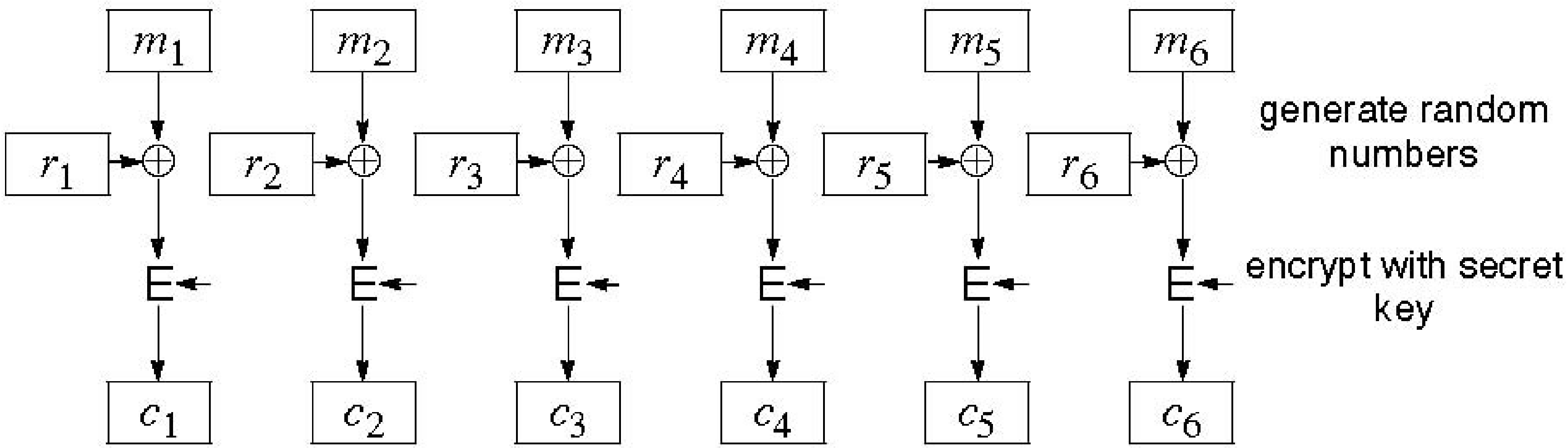
- Problem:
  - Identical Input➔ Identical Output
    - Deterministic: the same data block gets encrypted the same way, reveals patterns of data when a data block repeats.

ECB critics

- Weakness: Replay/Manipulation attack
  - Can insert encoded blocks
  - Reordering ciphertext results in reordered plaintext.
- Strength:
  - Errors in one ciphertext block do not propagate.
- Usage:
  - not recommended to encrypt more than one block of data
  - Encryption in database

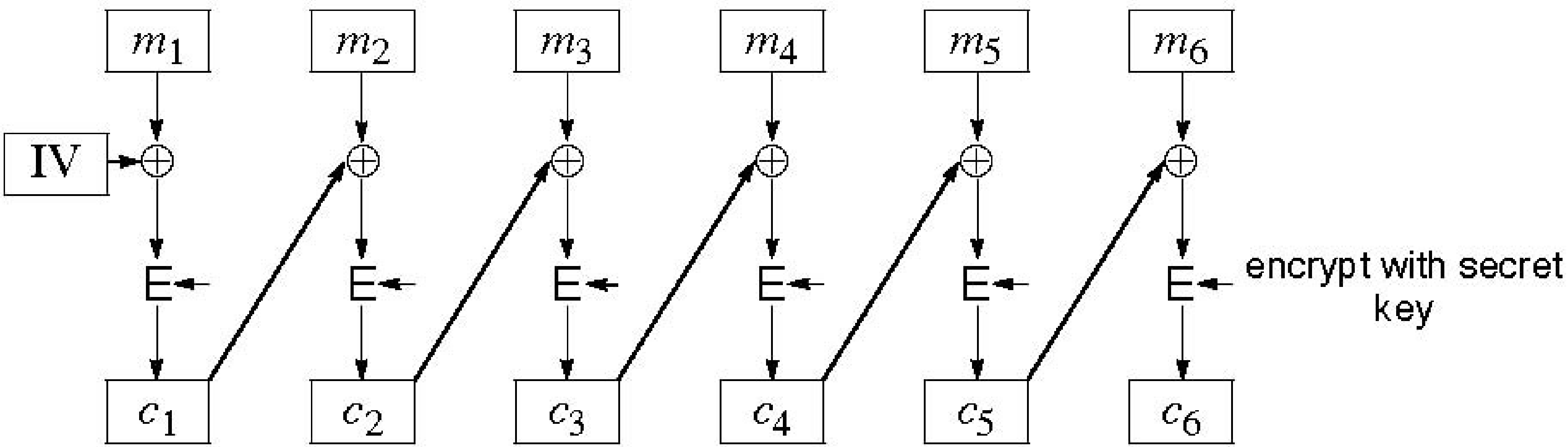
Cipher Block Chaining (CBC)

- Improving on ECB: think of adding a random number before encoding



CBC (cont.)

- The main idea:
  - Use C_i as random number block operation for i+1
  - So, need a so called Initial Value (IV)
    - If no IV, then one can guess changed blocks

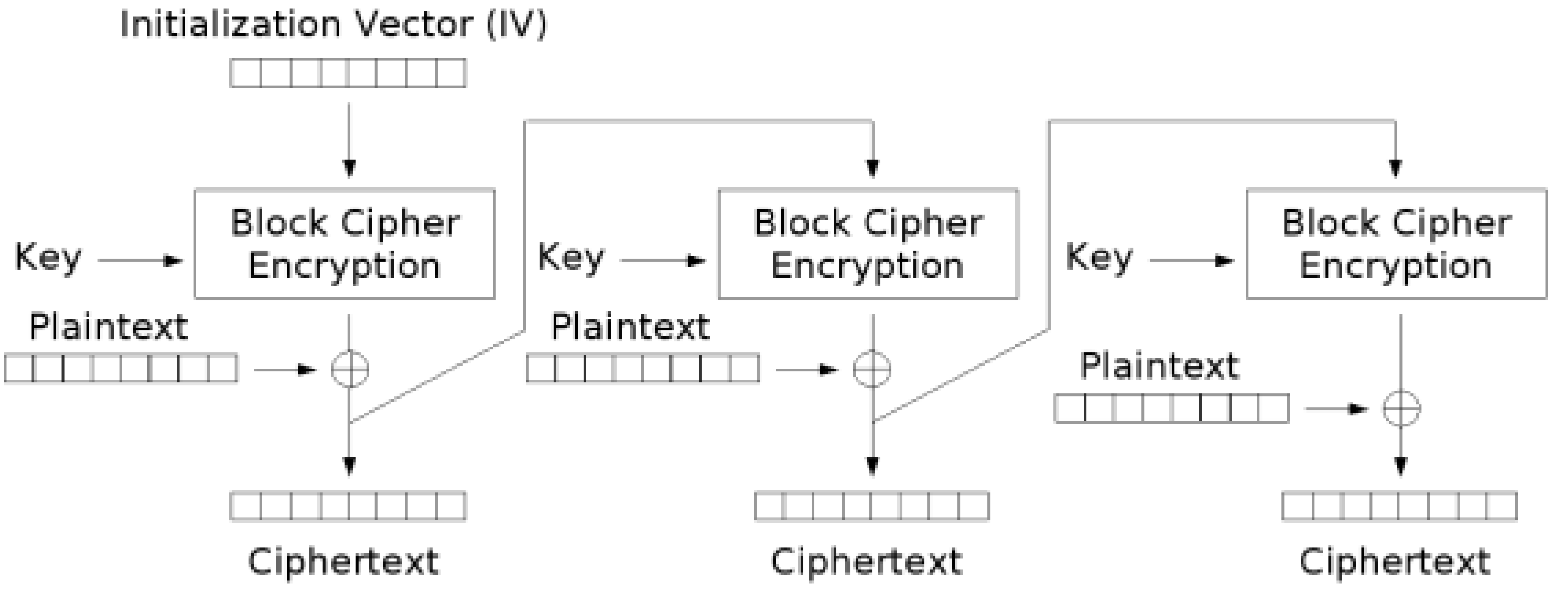


CBC critics

- Good
    - Randomized encryption: repeated text gets mapped to different encrypted data.
        - Can be proven to be “secure” assuming that the block cipher has desirable properties and that random IV’s are used
    - A ciphertext block depends on all preceding plaintext blocks
        - reorder affects decryption
- Bad
    - Errors in one block propagate to two blocks
        - one bit error in C_j affects all bits in M_j and one bit in M_j+1
    - Sequential encryption, cannot use parallel hardware
    - Observation: if $C_i = C_j$ then $E_k(M_i \oplus C_{i-1}) = E_k(M_j \oplus C_{j-1})$; thus $M_i \oplus C_{i-1} = M_j \oplus C_{j-1}$; thus $M_i \oplus M_j = C_{i-1} \oplus C_{j-1}$

Cipher Feedback (CFB)

- The message is XORed with the feedback of encrypting the previous block



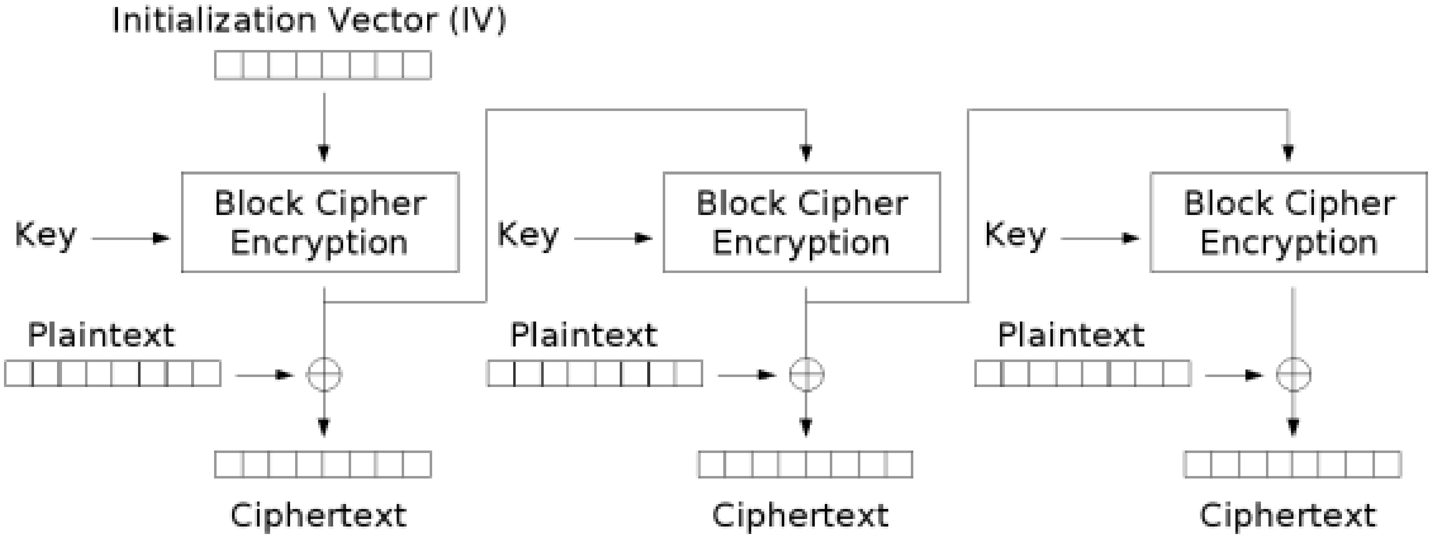
Cipher Feedback (CFB) mode encryption

CFB critics

- Good
  - Randomized encryption
  - A ciphertext block depends on all preceding plaintext blocks; reorder affects decryption
- Bad
  - Errors propagate for several blocks after the error, but the mode is self-synchronizing (like CBC).
  - Decreased throughput.
    - Can vary the number of bits feed back, trading off throughput for ease of use
  - Sequential encryption

Output Feedback (OFB)

- IV is used to generate a stream of blocks
- Stream is used a one-time pad and XOR'ed to plain text



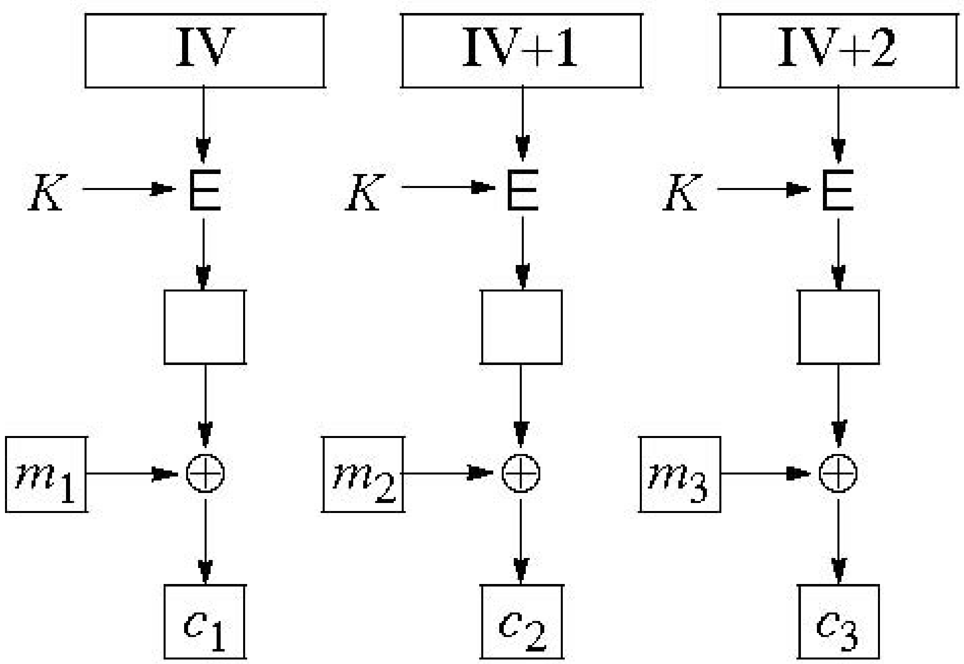
Output Feedback (OFB) mode encryption

OFB critics

- Randomized encryption
- Sequential encryption, but preprocessing possible
- Error propagation limited
- Subject to limitation of stream cipher

Counter Mode (CTR)

- If the same IV and key is used again,
  - XOR of two encrypted messages = XOR of plain text
- IV is incremented and used to generated one-time pad



CTR critics

- Software and hardware efficiency: different blocks can be encrypted in parallel.
- Preprocessing: the encryption part can be done offline and when the message is known, just do the XOR.
- Random access: decryption of a block can be done in random order, very useful for hard-disk encryption.
- Messages of arbitrary length: ciphertext is the same length with the plaintext (i.e., no IV).