CRYE 6127 Introduction to Cryptology

Jorge L. Villar

UBa Cyber Crypto Center, Fall 2025

Symmetric Key Encryption

Outline

Construction Strategies

- **Stream ciphers:** For arbitrarily long messages (e.g., data streams).
  A key $k \in \mathcal{K}$ and a nonce r are expanded to a **pseudorandom** stream, $g(r, k)$, used as a one-time pad.

$$\text{Enc}(k, m) = (r, m \oplus g(r, k))$$

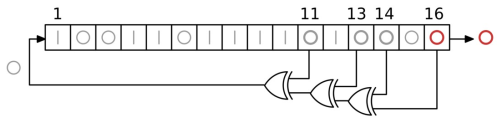
- **Block ciphers:** For messages with a fixed length.
  A **pseudorandom** permutation $f_k : \mathcal{M}_\ell \to \mathcal{C}_\ell$, used as a secret key, is applied to the message.

$$\text{Enc}(k, m) = f_k(m)$$

  with a **chaining mode** can encrypt arbitrarily long messages

Examples of stream ciphers

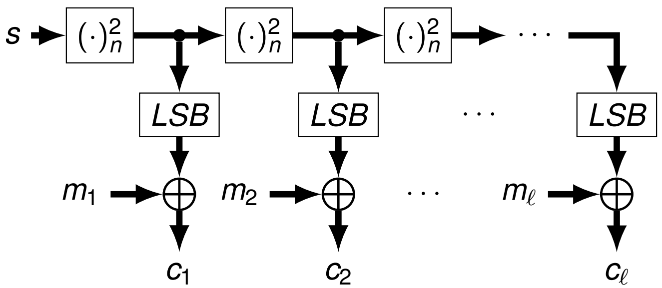
- **Linear Feedback Shift Registers.** Ultra fast but insecure.



(image based on https://commons.wikimedia.org/w/index.php?curid=60225898)

- **Blum-Blum-Shub generator.** Very slow. Provably secure.
- **Non-Linear Feedback Shift Registers.** Very fast. Widely used (GSM, Bluetooth, . . .). In general, they lead to weak symmetric encryption schemes.
- **Other designs:** RC4 (obsolete), chacha, ...

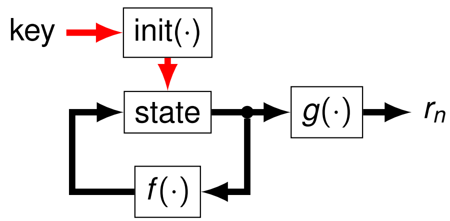
Stream Cipher from Blum-Blum-Shub Generator



$n = pq$, $p = 2p' + 1$, $q = 2q' + 1$
p, q, p', q' primes (512 to 1024 bits).

RC4 Stream Cipher

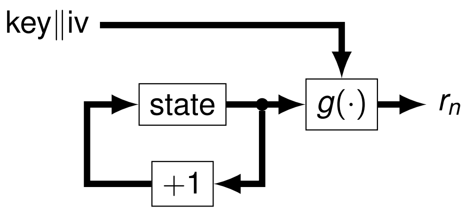
- Internal state: a permutation s of $\{0, \dots, 255\}$ and two counters i, j.



- initial state: computed from the key.
- Every message byte m_n is XORed with the output byte r_n.
- The key is too short (from 40 bits to 128 bits).
- No specified way to use an initialization value (iv).
- Some real systems using RC4 have been attacked.

Chacha Stream Cipher

- Internal state: just a counter i.
- It works with 512 bit words.



- initial state: $i = 0$.
- Every message word m_n is XORed with the output word r_n.
- key has 256 bits and the iv has 64 bits.
- fully parallelizable (random access to words)
- No successful attack is currently known.

Outline

Practical Construction of Block Ciphers

Heuristic constructions based on:

- **Confusion:** every bit in the ciphertext must depend on several bits of the key.
- **Diffusion:** flipping a single plaintext bit must change half of the ciphertext bits.
- **Iteration:** the encryption procedure consists of a given number of rounds.

Structure:

- **Key expansion:** a set of round keys is generated from the original key
- **Mixing** of permutation (shuffling, shifting. . .), substitution (S-boxes) and XORing the subkeys in each round.

Example: DES

Key size: 56 bits

Message and ciphertext block size: 64 bits

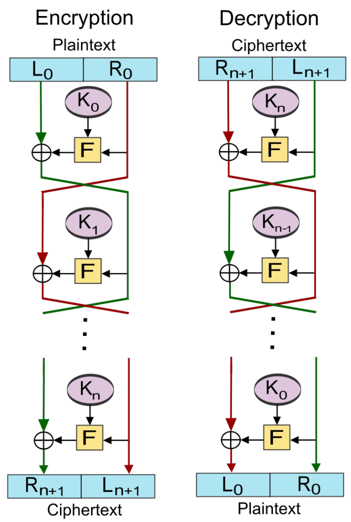
Number of encryption rounds: 16

Key expansion: from 56 bits to 768 bits (16 subkeys, each one of 48 bits)

Encryption rounds: Feistel Network

- Start with the 64 bit message block
- In each iteration, divide the intermediate block into two halves (L_i, R_i)
- Apply a transformation to compute next block $(L_{i+1}, R_{i+1}) = (R_i, L_i \oplus F(k_i, R_i))$
- The resulting ciphertext block is (R_{16}, L_{16}).

Feistel Network



Example: DES (II)

The function F combines:

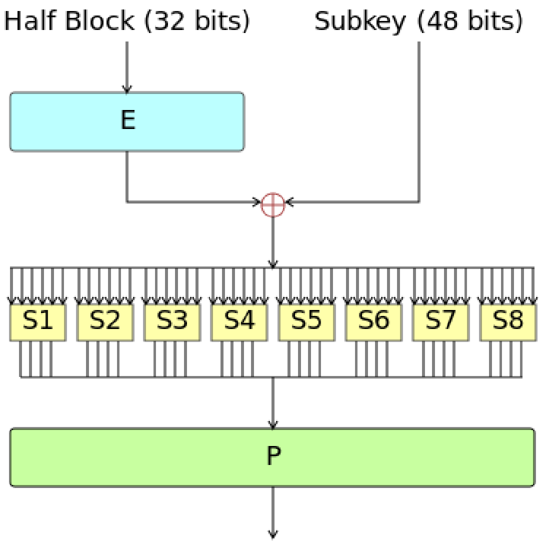
- Expands the 32 bit half-block into 48 bits, and XORs the round subkey.
- Divides the 48 bits into 8 6-bit words.
- Transforms each word into a 4 bit word with a table (S-box).
- Glues the 4 bit words to obtain a 32 bit half-block.
- Applies a fixed premutation to the 32 bits.

3DES combines three independent keys (168 bits in total) and it is still considered secure.

$3DES_{k_1,k_2,k_3}(m) = DES_{k_3} \circ DES^{-1}_{k_2} \circ DES_{k_1}(m)$

More details in
http://en.wikipedia.org/wiki/Data_Encryption_Standard

Example: DES Round Function



Example: AES256

Key size: 256 bits

Message and ciphertext block size: 128 bits (4×4 byte matrix)

Number of encryption rounds: 14

Key expansion: from 256 bits to 1920 bits (15 subkeys, each one a 4×4 matrix of bytes) in an iterative procedure (it adds 4 new bytes in each iteration)

Encryption round:

- Apply a substitution box to each byte
- Perform rotation operations to rows
- Perform linear transformation to columns
- XOR with the round key

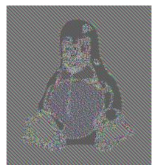
More details in
http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Outline

Block Chaining Modes

A strategy to encrypt messages larger than one block: Partition $m \in \{0,1\}^{n\ell}$ into n blocks $m_1, \ldots, m_n \in \{0,1\}^\ell$ and encrypt each block separately.

ECB (Electronic Codebook):
$c_i = \mathrm{Enc}(k, m_i)$



(from Wikipedia)

Equal message blocks result in equal ciphertext blocks!

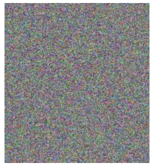
Block Chaining Modes

A strategy to encrypt messages larger than one block: Partition $m \in \{0,1\}^{n\ell}$ into n blocks $m_1, \dots, m_n \in \{0,1\}^{\ell}$ and encrypt each block separately.

CBC (Cipher Block Chaining):

$c_0 = iv$

$c_i = \mathrm{Enc}(k, m_i \oplus c_{i-1})$

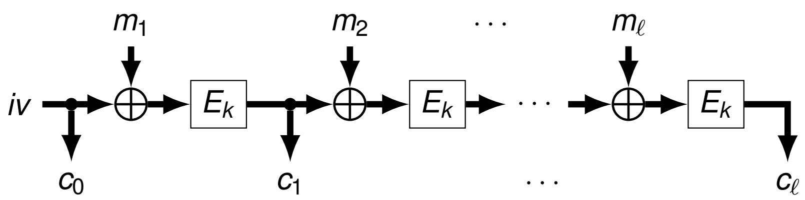


(from Wikipedia)

The ciphertext has an extra random block c_0, the "initialization vector"

CBC Mode

Comp...

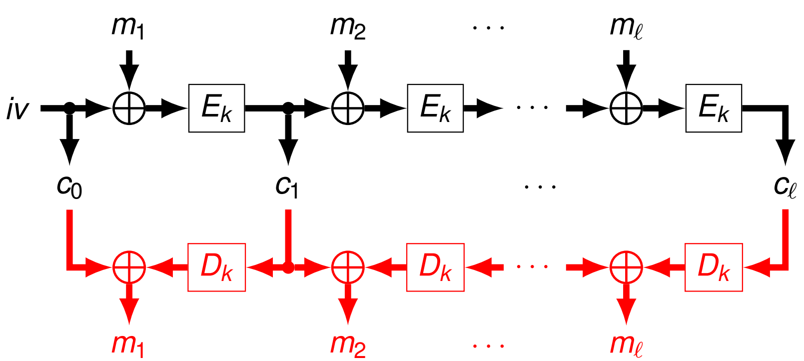


$c_i = c_j \quad \Rightarrow \quad m_i \oplus c_{i-1} = m_j \oplus c_{j-1} \quad \Rightarrow \quad m_i \oplus m_j$ revealed!

Collision probability must be small

CBC Mode

Comp...



$c_i = c_j \quad \Rightarrow \quad m_i \oplus c_{i-1} = m_j \oplus c_{j-1} \quad \Rightarrow \quad m_i \oplus m_j$ revealed!

Collision probability must be small

Other Block Chaining Modes

- CFB (Cipher Feedback) operation mode:

  $c_0 = iv$

  $c_i = \mathrm{Enc}(k, c_{i-1}) \oplus m_i$

- OFB (Output Feedback) operation mode:

  $r_0 = iv$

  $r_i = \mathrm{Enc}(k, r_{i-1})$

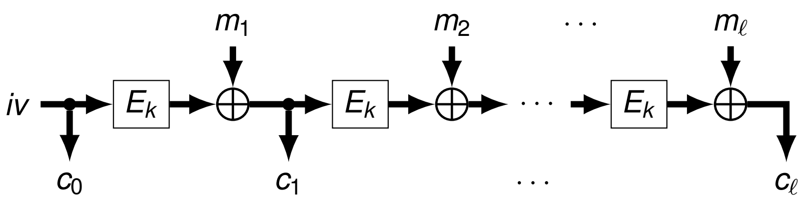
  $c_i = m_i \oplus r_i$

- CTR (Counter) operation mode:

  $r_i = \mathrm{Enc}(k, iv + i)$

  $c_i = m_i \oplus r_i$

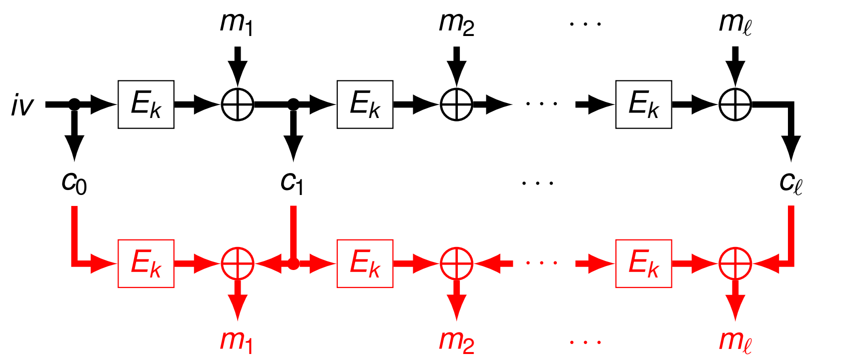
CFB Mode



$c_{i-1} = c_{j-1} \quad \Rightarrow \quad m_i \oplus c_i = m_j \oplus c_j \quad \Rightarrow \quad m_i \oplus m_j$ revealed!

Collision probability must be small

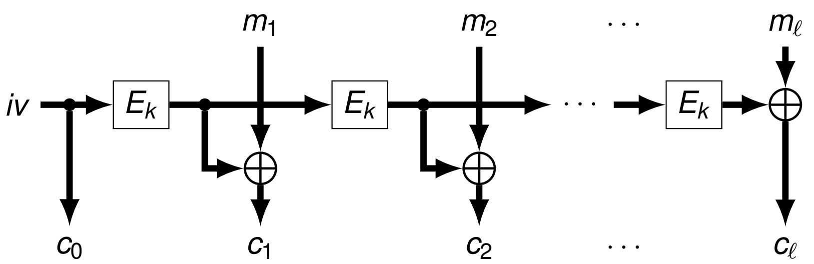
CFB Mode



$c_{i-1} = c_{j-1} \quad \Rightarrow \quad m_i \oplus c_i = m_j \oplus c_j \quad \Rightarrow \quad m_i \oplus m_j$ revealed!

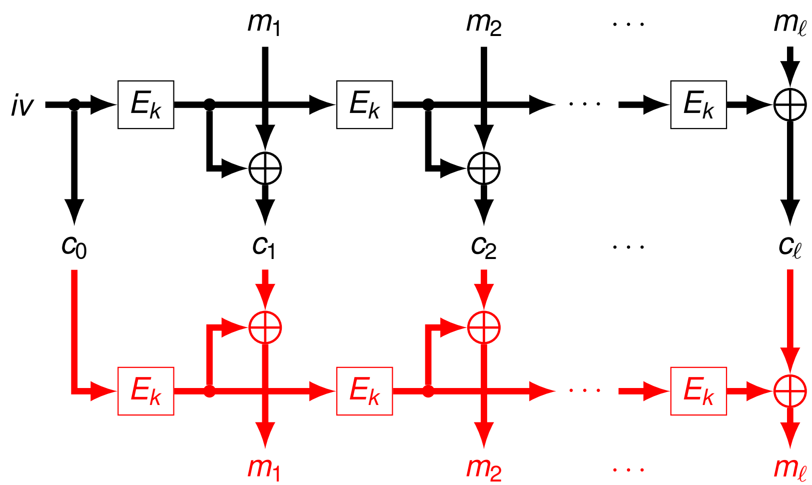
Collision probability must be small

OFB Mode



A short cycle of E_k of length r reveals $m_i \oplus m_{i+r}$ for all i

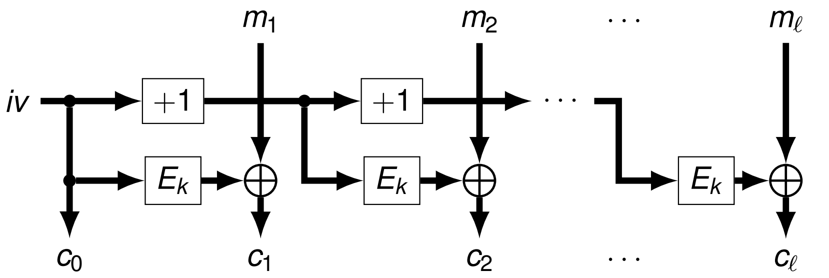
OFB Mode



A short cycle of E_k of length r reveals $m_i \oplus m_{i+r}$ for all i

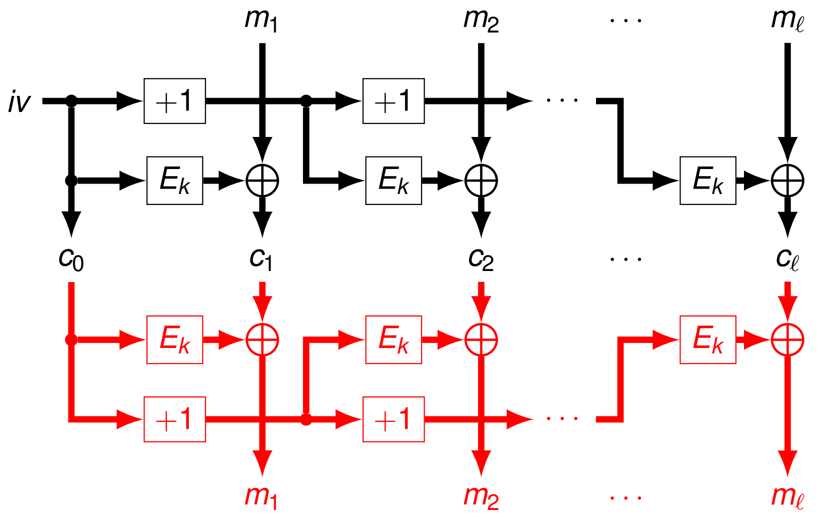
CTR Mode

Comp...



CTR Mode

Comp...



Comparison of Block Chaining Modes

- CFB, OFB and CTR require only the encryption box (even for decryption)
- CBC and CFB are “self-synchronizable”
- CTR can use random access for encryption and decryption (i.e., it is parallelizable)
- CBC and CFB decryption are also parallelizable (random access)
- A corrupted *iv* in OFB and CTR makes all blocks undecryptable
- Only ECB and CBC require padding of the last incomplete message block

CBC... CFB... OFB... CTR...

Plaintext Padding

If the last message block is not complete, a bit string is appended such that:

- It has the minimum possible length.
- It has to not introduce any ambiguity in decryption.

Example:

- Append a string: “1” followed by zero or more “0”.
- The empty string is not a valid padding. A last complete block also needs to be padded!
- In decryption, the last “1” and all trailing zeros (if any) are discarded.

Outline

1. Stream Ciphers
2. Block Ciphers
3. Block Chaining Modes
4. Key Management

Key Generation and Storage Issues

Key generation needs truly random bits (hard to extract from nature)

- Low entropy sources (e.g., passwords) are not enough.
- Smoothing and entropy accumulation (heuristic techniques: diffusion, iteration), or dedicated hardware.
- Known attacks exploiting the lack of true randomness.

Keys have to be stored securely (Then, new keys required)

- Use hierarchical encryption: encrypt some keys under a common dedicated key.
- Can incur in **circularity**: plaintext depending on the key!
- Use a **Key Derivation Function** applied to a passphrase and a (public) nonce.

CRYE 6127 Introduction to Cryptology

Jorge L. Villar

UBa Cyber Crypto Center, Fall 2025

Symmetric Key Encryption

—END—