Block ciphers

Oleksandr Kazymyrov

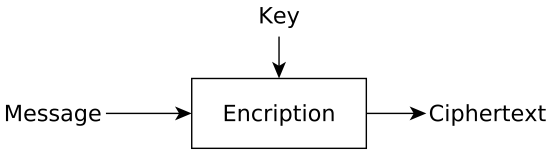
University of Bergen,
Norway

November 10, 2011

Outline

Definitions

The block cipher (BC) encrypts a block of plaintext or message M into a block of ciphertext C using a secret key K.



Definitions

Let

$$E : \{0,1\}^l \times \{0,1\}^k \mapsto \{0,1\}^l$$

be a function taking a key K of length k bits and input M of length l to return output $E(M,K)$. For each key K let $E_K : \{0,1\}^l \mapsto \{0,1\}^l$ be the function defined by

$$E_K(M) = E(M,K)$$

E is a block cipher if

- $E_K : \{0,1\}^l \mapsto \{0,1\}^l$ is a permutation for every K, i.e. it has an inverse E_K^{-1},
- E_K, E_K^{-1} are efficiently computable,

where $E^{-1}(M,K) = E_K^{-1}(M)$.

Block vs Stream Ciphers

	Block Ciphers	Stream Ciphers
Process messages	by blocks	by bit or byte
Maximum message length	depends on the encryption mode	limited
Performance	fast	extremely fast
Usually usage	software	hardware

General structure of BC

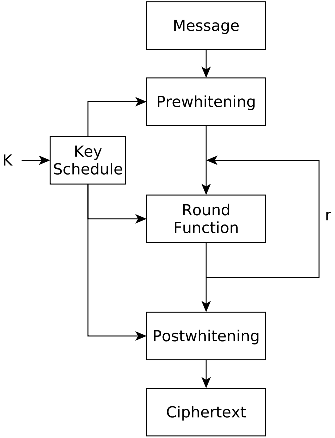
Most of block ciphers are constructed by repeating of simple functions. This approach is known as iterated block cipher. Each iteration is named a round, and the repeated function - the round function.

Prewhitening - the initial transformation is applied to input message

Postwhitening - the final transformation is applied to output of round function

Key Schedule - function of generation subkeys from a master key

r - number of rounds



Prewhitening and postwhitening

Prewhitening and postwhitening should be simpler and much faster then round function. This approach makes cryptanalysis more difficult.

Prewhitening and postwhitening can be:

- missing
- implemented as an extra addition with a key
- based on the round function
- presented as individual functions which are not like the round function

Round function

Round function usually consists of linear (P-box) and nonlinear layers (S-box).

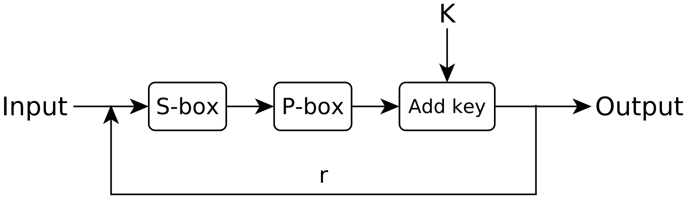


Figure: Round function

Key schedule

A key schedule is an algorithm that is based on the master key and calculates the subkeys for all stages of encryption.
Key schedule of block ciphers can be:

- missing (subkeys are the part of master key)
- based on trivial linear transformation
- based on round function (is used linear and nonlinear layers)
- constructed taking into account new types of attacks

Outline

Feistel network

- Splitting plaintext block into left and right halves

$$P = (L_0, R_0)$$

- For each round $i = 1, 2, \ldots, r$ compute

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

Where F is round function and K_i - subkey.

- Ciphertext

$$C = (L_r, R_r)$$

Feistel network

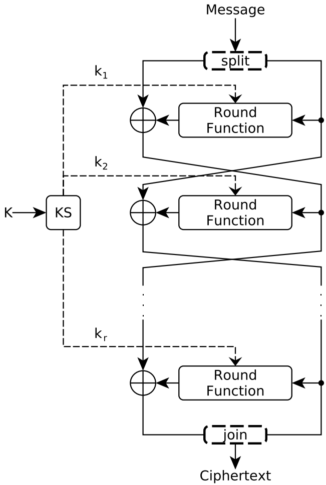
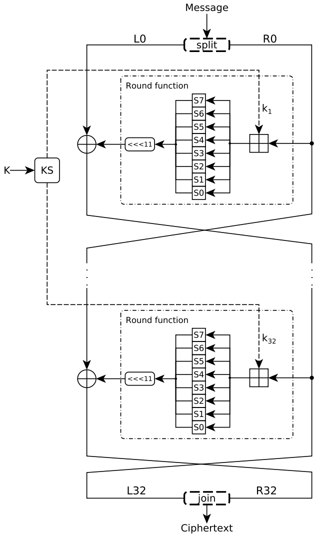


Figure: Feistel network

GOST

GOST parameters

- block size: 64 bits
- key length: 256 bit
- number of rounds: 32



Substitution-Permutation Network

SPN structure

$$E_K = \bigcirc_{i=1}^{r} \left(\sigma[K^i] \circ \tau \circ \gamma\right) \circ \sigma[K^0]$$

γ - a nonlinear layer (S-box),
τ - a linear layer (P-box),
$\sigma[K^i]$ - addition with K_i.
Confusion: the ciphertext statistics should depend on the plaintext statistics in a manner too complicated to be exploited by the cryptanalyst.
Diffusion: each digit of the plaintext and each digit of the secret key should influence many digits of the ciphertext.

Substitution-Permutation Network

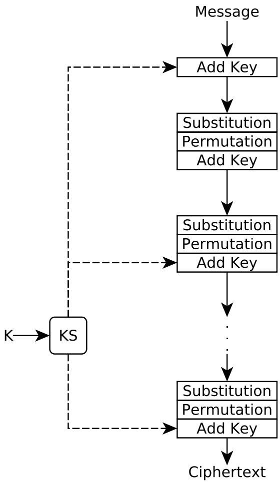


Figure: SPN

AES

AES encryption procedure

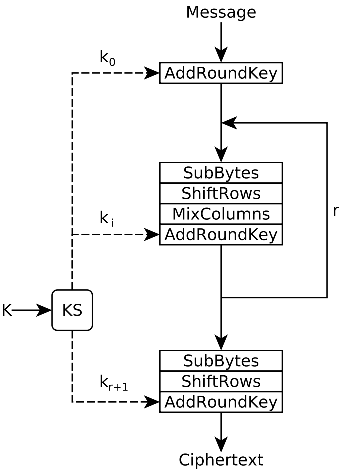
$$E_K = \sigma[K^{r+1}] \circ \tau \circ \gamma \bigcirc_{i=1}^{r} \left(\sigma[K^i] \circ \theta \circ \tau \circ \gamma\right) \circ \sigma[K^0]$$

γ - a nonlinear substitution function where each byte is replaced with another according to a lookup table (SubBytes),
τ - a transposition function where each row of the state is shifted cyclically a certain number of steps (ShiftRows),
θ - a mixing operation which operates on the columns of the state, combining the four bytes in each column (MixColumns),
$\sigma[K^i]$ - addition with K_i modulo 2 (AddRoundKey).

AES

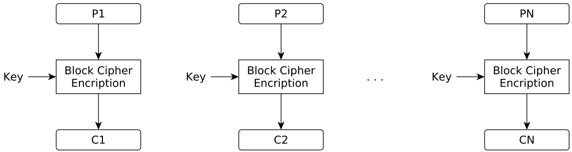
AES parameters

- block size: 128 bits
- key length: 128, 192, 256 bit
- number of rounds: 10, 12, 14

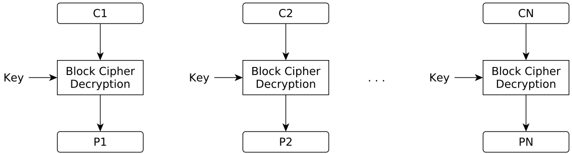


Outline

Electronic Codebook Mode (ECB)



Encryption procedure



Decryption procedure

Advantages and Limitations of ECB

- repetitions in message can be seen in ciphertext
- weakness due to encrypted message blocks being independent
- main use is to send a few blocks of data.

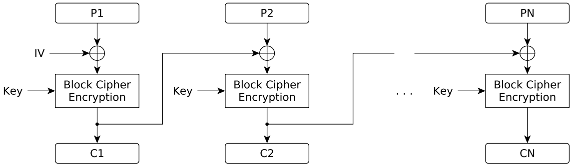
Advantages and Limitations of ECB

- repetitions in message can be seen in ciphertext
- weakness due to encrypted message blocks being independent
- main use is to send a few blocks of data.

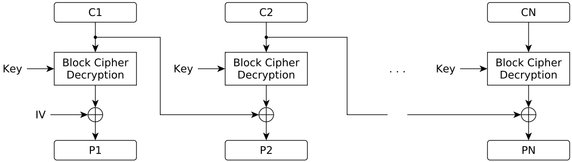
Advantages and Limitations of ECB

- repetitions in message can be seen in ciphertext
- weakness due to encrypted message blocks being independent
- main use is to send a few blocks of data.

Cipher Block Chaining Mode (CBC)



Encryption procedure



Decryption procedure

Advantages and Limitations of CBC

- each ciphertext block depends on all message blocks
- some changes in the message affects all next ciphertext blocks
- Initial Value (IV) is needed to be known to sender and receiver

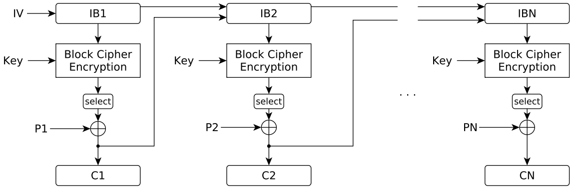
Advantages and Limitations of CBC

- each ciphertext block depends on all message blocks
- some changes in the message affects all next ciphertext blocks
- Initial Value (IV) is needed to be known to sender and receiver

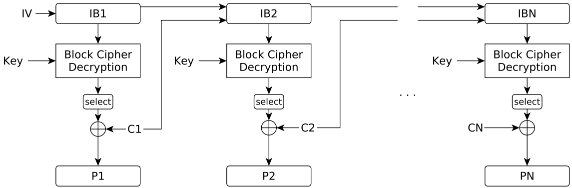
Advantages and Limitations of CBC

- each ciphertext block depends on all message blocks
- some changes in the message affects all next ciphertext blocks
- Initial Value (IV) is needed to be known to sender and receiver

Cipher Feedback Mode (CFB)



Encryption procedure



Decryption procedure

Advantages and Limitations of CFB

- appropriate receiving of data
- changing of IV is needed after every n-bits of encryption
- errors propagate for several blocks after the error

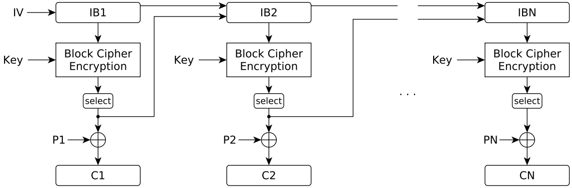
Advantages and Limitations of CFB

- appropriate receiving of data
- changing of IV is needed after every n-bits of encryption
- errors propagate for several blocks after the error

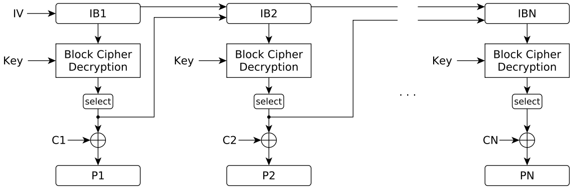
Advantages and Limitations of CFB

- appropriate receiving of data
- changing of IV is needed after every n-bits of encryption
- errors propagate for several blocks after the error

Output Feedback Mode (OFB)



Encryption procedure



Decryption procedure

Advantages and Limitations of OFB

- is used before the message is available
- sender and receiver must remain in sync, and some recovery method is needed to ensure this occurs
- never use the same sequence (key+IV)

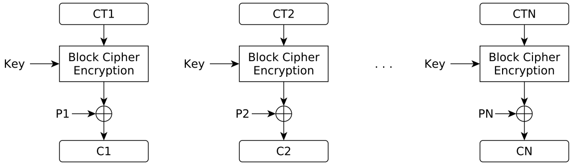
Advantages and Limitations of OFB

- is used before the message is available
- sender and receiver must remain in sync, and some recovery method is needed to ensure this occurs
- never use the same sequence (key+IV)

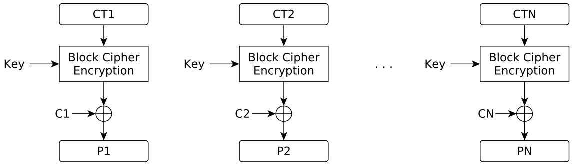
Advantages and Limitations of OFB

- is used before the message is available
- sender and receiver must remain in sync, and some recovery method is needed to ensure this occurs
- never use the same sequence (key+IV)

Counter Mode (CTR)



Encryption procedure



Decryption procedure

Advantages and Limitations of CTR

- efficiency
  - can do parallel encryptions
  - in advance of need
  - good for high speed links
- random access to encrypted data blocks
- provable security, based on security of cipher
- must ensure that key/counter values will never be reused

Advantages and Limitations of CTR

- efficiency
  - can do parallel encryptions
  - in advance of need
  - good for high speed links
- random access to encrypted data blocks
- provable security, based on security of cipher
- must ensure that key/counter values will never be reused

Advantages and Limitations of CTR

- efficiency
  - can do parallel encryptions
  - in advance of need
  - good for high speed links
- random access to encrypted data blocks
- provable security, based on security of cipher
- must ensure that key/counter values will never be reused

Advantages and Limitations of CTR

- efficiency
  - can do parallel encryptions
  - in advance of need
  - good for high speed links
- random access to encrypted data blocks
- provable security, based on security of cipher
- must ensure that key/counter values will never be reused

Outline

Padding methods

- zero padding
- one bit padding
- ISO 10126
- ANSI X.923
- PKCS7 (RFC 5652)
- Method 3 of ISO/IEC 9797-1

Padding methods

- zero padding

$$\mid x_0x_1x_2x_4 \mid \; \ldots \; \mid x_{n-4}x_{n-3}x_{n-2}x_{n-1} \mid x_n \; \mathbf{0\ 0\ 0} \mid$$

- one bit padding

$$\mid x_0x_1x_2x_4 \mid \; \ldots \; \mid x_{n-4}x_{n-3}x_{n-2}x_{n-1} \mid x_n \; \mathbf{1\ 0\ 0} \mid$$

- ISO 10126

$$\mid x_0x_1x_2x_4 \mid \; \ldots \; \mid x_{n-4}x_{n-3}x_{n-2}x_{n-1} \mid x_n \; \textbf{FA 10 5B} \mid$$

Padding methods

- ANSI X.923

$$| x_0x_1x_2x_4 | \ldots | x_{n-4}x_{n-3}x_{n-2}x_{n-1} | x_n \textbf{ 00 00 03 } |$$

- PKCS7 (RFC 5652)

$$| x_0x_1x_2x_4 | \ldots | x_{n-5}x_{n-4}x_{n-3}x_{n-2} | x_{n-1}x_n \textbf{ 02 02 } |$$

- Method 3 of ISO/IEC 9797-1

$$| \textbf{N} | x_0x_1x_2x_4 | \ldots | x_{n-4}x_{n-3}x_{n-2}x_{n-1} | x_n \textbf{ 00 00 00 } |$$