Block ciphers

Tanja Lange

Eindhoven University of Technology

2WF80: Introduction to Cryptology

Block cipher

- Encrypts n bits of message to n bits of ciphertext using ℓ-bit key.

$$\mathsf{Enc} : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^n, \quad \mathsf{Enc}_k(m) = c.$$

- Encryption is invertible with $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$.
- Shannon's design goals:
  - confusion: bits get mixed;
  - diffusion: differences spread out.
- Messages longer than one block have to be split into blocks. See video Modes of operation for details and padding.
- Do *not* just encrypt blockwise!

Block cipher

- Encrypts n bits of message to n bits of ciphertext using ℓ-bit key.

$$\mathrm{Enc} : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^n, \quad \mathrm{Enc}_k(m) = c.$$

- Encryption is invertible with $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$.
- Shannon's design goals:
  - confusion: bits get mixed;
  - diffusion: differences spread out.
- Messages longer than one block have to be split into blocks.
  See video Modes of operation
  for details and padding.
- Do *not* just encrypt blockwise!
  Frequency analysis works same as
  for substitution cipher.
- Remember the ECB penguin
  as warning not to use
  electronic codebook mode.

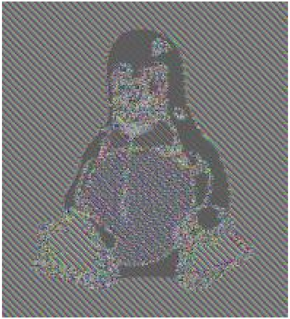


Image credit: By en:User:Lunkwill

Inside the block cipher: Feistel network

Named after Feistel (IBM); used in Lucifer design.

Splits message into two halves, uses function on right half to encrypt left half; then swaps sides.

Typically want an even number of rounds so that both halves are encrypted equally often.

Functions f_i use (pieces of) the secret key.

Don't need f_i to be invertible:

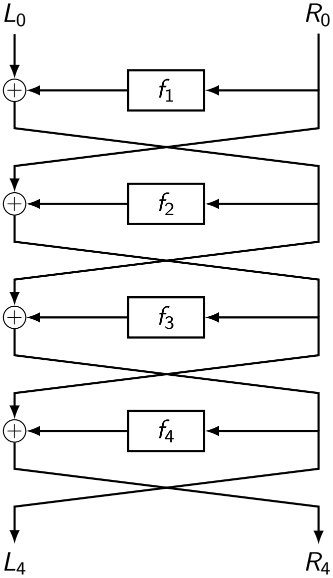


Image credit: Jérémy Jean

Inside the block cipher: Feistel network

Named after Feistel (IBM);
used in Lucifer design.

Splits message into two halves,
uses function on right half
to encrypt left half;
then swaps sides.

Typically want an even number
of rounds so that both halves
are encrypted equally often.

Functions f_i use (pieces of)
the secret key.

Don't need f_i to be invertible:

$R_3 = L_4$ (part of output)
$L_3 = R_4 + f_4(R_3)$ (computable).

Repeat till (L_0, R_0) is recovered.
Great flexibility to build f_i.

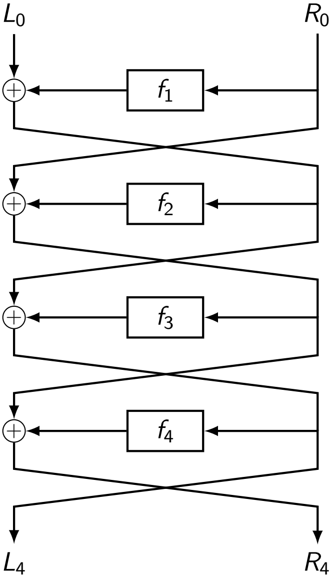
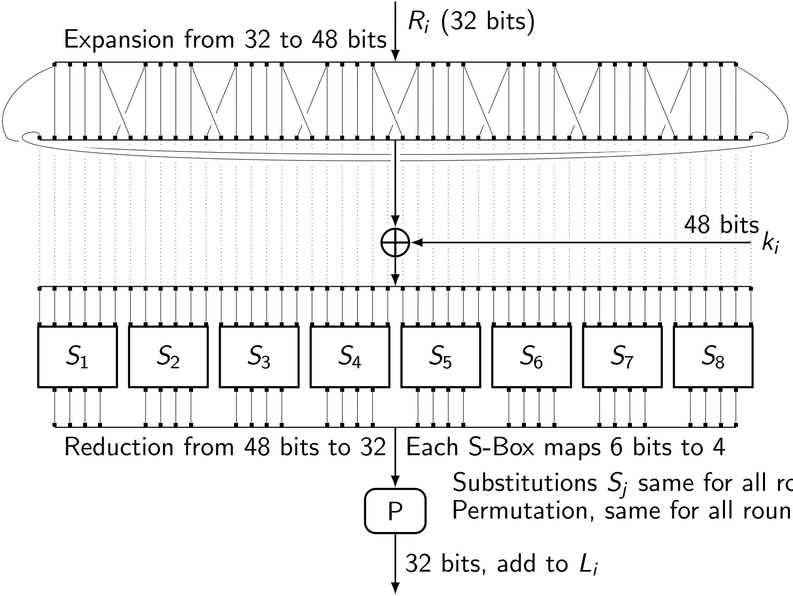


Image credit: Jérémy Jean

Function f_i (rotated by 90 degrees) for DES



Substitutions S_j same for all rounds
Permutation, same for all rounds

Image credit: adapted from Roberto Avanzi

Data Encryption Standard (DES)

- Expansion (32 to 48 bits) and compression (6 to 4 bits) are meant to amplify differences.
- S-boxes (Substitution boxes) are nonlinear, given by lookup tables. $S_i(x) + S_i(1) \neq S_i(x + i)$

Data Encryption Standard (DES)

- Expansion (32 to 48 bits) and compression (6 to 4 bits) are meant to amplify differences.
- S-boxes (Substitution boxes) are nonlinear, given by lookup tables. $S_i(x) + S_i(1) \neq S_i(x + i)$
- No design rationale offered for S-boxes.
- S-boxes were changed by NSA from original IBM design.
- 1990: Biham and Shamir develop differential cryptanalysis. DES S-boxes are stronger against this than original IBM ones.

Data Encryption Standard (DES)

- Expansion (32 to 48 bits) and compression (6 to 4 bits) are meant to amplify differences.
- S-boxes (Substitution boxes) are nonlinear, given by lookup tables. $S_i(x) + S_i(1) \neq S_i(x+i)$
- No design rationale offered for S-boxes.
- S-boxes were changed by NSA from original IBM design.
- 1990: Biham and Shamir develop differential cryptanalysis. DES S-boxes are stronger against this than original IBM ones.
- However, the key has only 56 bits.
- Key size was obviously too small – IBM proposal had 128 bits.
  - 1976 Diffie and Hellman raise alarm about key size.
  - 1998 "DES cracker" by EFF breaks DES encryption by brute-force key search on 250k USD custom-built device.
  - 2005 DES is officially withdrawn by NIST (National Institute for Standards and Technology).
  - 2006 COPACOBANA (FPGA cluster by Ruhr University Bochum) "How to Break DES for 8,980 EUR"
- DES is still around – mostly in the financial industry; (weak) justification: Hardware Security Modules (HSMs) are expensive.

Other block ciphers

- If DES is still used then as 3-DES: $c = \mathrm{Enc}_{k_3}(\mathrm{Dec}_{k_2}(\mathrm{Enc}_{k_1}(m)))$.
- This computes DES for $k_1 = k_2 = k_3$.
- For 3 different keys attack cost is lower than $2^{3 \cdot 56}$:

Other block ciphers

- If DES is still used then as 3-DES: $c = \mathrm{Enc}_{k_3}(\mathrm{Dec}_{k_2}(\mathrm{Enc}_{k_1}(m)))$.
- This computes DES for $k_1 = k_2 = k_3$.
- For 3 different keys attack cost is lower than $2^{3\cdot 56}$:
  Attack given pair (m, c):
  Make table of $\mathrm{Dec}_{\bar{k}_3}(c)$ for all 2^{56} keys $\bar{k}_3$, find match with $\mathrm{Dec}_{\bar{k}_2}(\mathrm{Enc}_{\bar{k}_1}(m))$ (running through all $\bar{k}_2$ and $\bar{k}_1$).

  This takes 2^{56} storage, 2^{112} time, not $2^{3\cdot 56}$ time.

Other block ciphers

- If DES is still used then as 3-DES: $c = \text{Enc}_{k_3}(\text{Dec}_{k_2}(\text{Enc}_{k_1}(m)))$.
- This computes DES for $k_1 = k_2 = k_3$.
- For 3 different keys attack cost is lower than $2^{3\cdot 56}$:
  Attack given pair (m, c):
  Make table of $\text{Dec}_{\bar{k}_3}(c)$ for all 2^{56} keys $\bar{k}_3$, find match with $\text{Dec}_{\bar{k}_2}(\text{Enc}_{\bar{k}_1}(m))$ (running through all $\bar{k}_2$ and $\bar{k}_1$).

  This takes 2^{56} storage, 2^{112} time, not $2^{3\cdot 56}$ time.
- 2001 New standard:
  AES (Advanced Encryption Standard) has block size 128 bits;
  keys of 128, 192, or 256 bits.
- AES was chosen in competition hosted by NIST.
- AES based on Rijndael by Daemen and Rijmen.
- AES is not based on Feistel cipher. Much more theory available after 40+ years of public research. Latest approach: sponges.
- Design elements of DES used in PRESENT lightweight cipher (uses single S-box; 80-bit key).