Recap: block ciphers

- **Goal**: encryption of a fixed length message (block)

Tool: strong pseudorandom permutation family (PRP)

- similar to PRF
- maps n-bit strings to n-bit strings **bijectively**
- for $k \in \{0,1\}^n$, $F_k(.)$ is a member of the family
- a random element of the family is PPT-indistinguishable from a random function using the functions and the inverses.

Recap: block cipher

Block cipher

Let F be a strong PRP

Gen $k \in_R \{0,1\}^n$

Enc for key $k \in \{0,1\}^n$ and message $m \in \{0,1\}^n$ and $r \in_R \{0,1\}^n$: let $c = Enc_k(m) = (r, F_k(r) \oplus m)$.

Dec for key k and ciphertext $c = (r, s)$, we have $Dec_k(r, s) = F_k(r) \oplus s$.

Feistel network

Basic idea

- Goal: to encrypt $m \in \{0,1\}^n$, n: block length
- $t+1$ rounds of iterative computation for encryption
- subkeys obtained from k for each round
- F: round function
- mix and change subblocks of the message

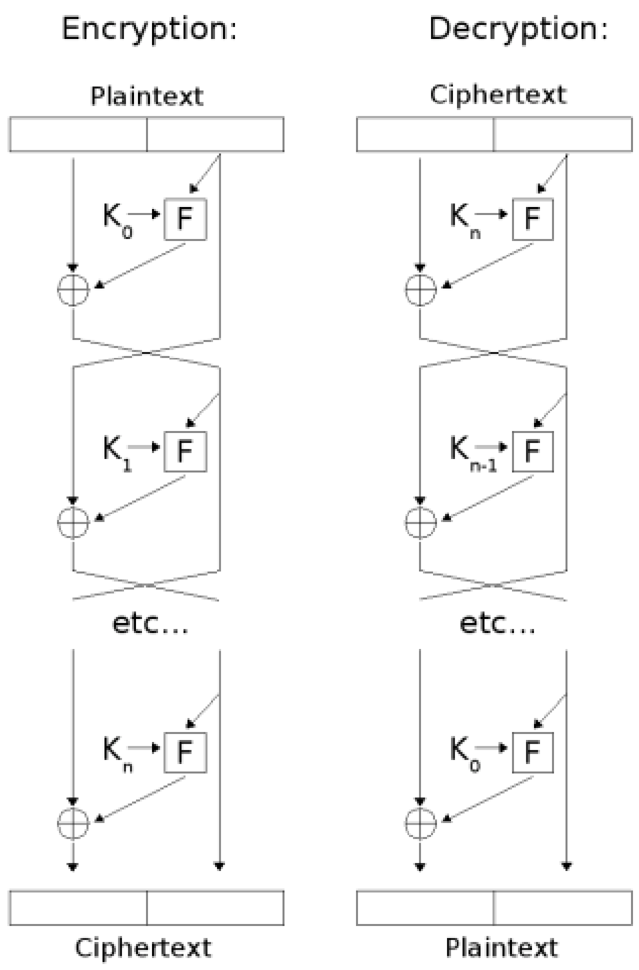
Feistel network

Feistel network

Gen $F : \{0,1\}^n \times \{0,1\}^n \Rightarrow \{0,1\}^n$ round-function, $k_0, k_1, \ldots, k_t$ subkeys

Enc $m = (L_0, R_0), i = 0, .., t+1 :$
$L_{i+1} = R_i, R_{i+1} = L_i \oplus (F_{k_i}(R_i))$
$c = (L_{t+1}, R_{t+1})$

Dec $i = n, n-1.., 0 : R_i = L_{i+1}, L_i = R_{i+1}(F_{k_i}(L_{i+1}))$
$m = (L_0, R_0)$

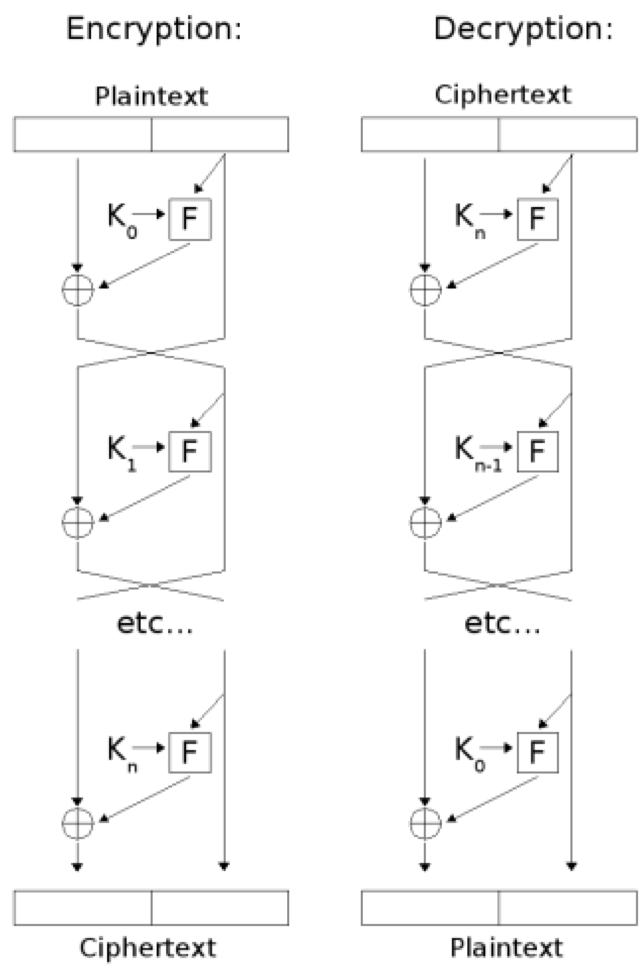


Feistel Cipher

Feistel networks

Properties

- if $F(.)$ is PRF $\Rightarrow$ PRP after 3 rounds, strong PRP after 4 rounds
- $F(.)$ any function (not necessarily invertible)
- unbalanced versions
- randomized ciphertext
- format preserving encryption
- GOST, DES, RC5, Blowfish, ...



Feistel Cipher

Feistel network

GOST

- block length: 64 bit, key size: 256 bit, 32 rounds
- round function: $F_{k_i}(m_i) : m_i \boxplus k_i \longrightarrow$ S-box $\longrightarrow <<< 11$
- 8 S-boxes of size 4x4

Properties

- soviet origins
- theoretical break
- practical break: 2^{192} time for 2^{64} pieces of data

Substitution-permutation networks

Basic idea

- encrypt $m \in \{0,1\}^n$, where n is the block lentgh
- several rounds
- subkeys generated from k
- S(ubst)-box and P(ermut)-box structure (changes with each round)
- S-box: substitution function
- P-box: permutation

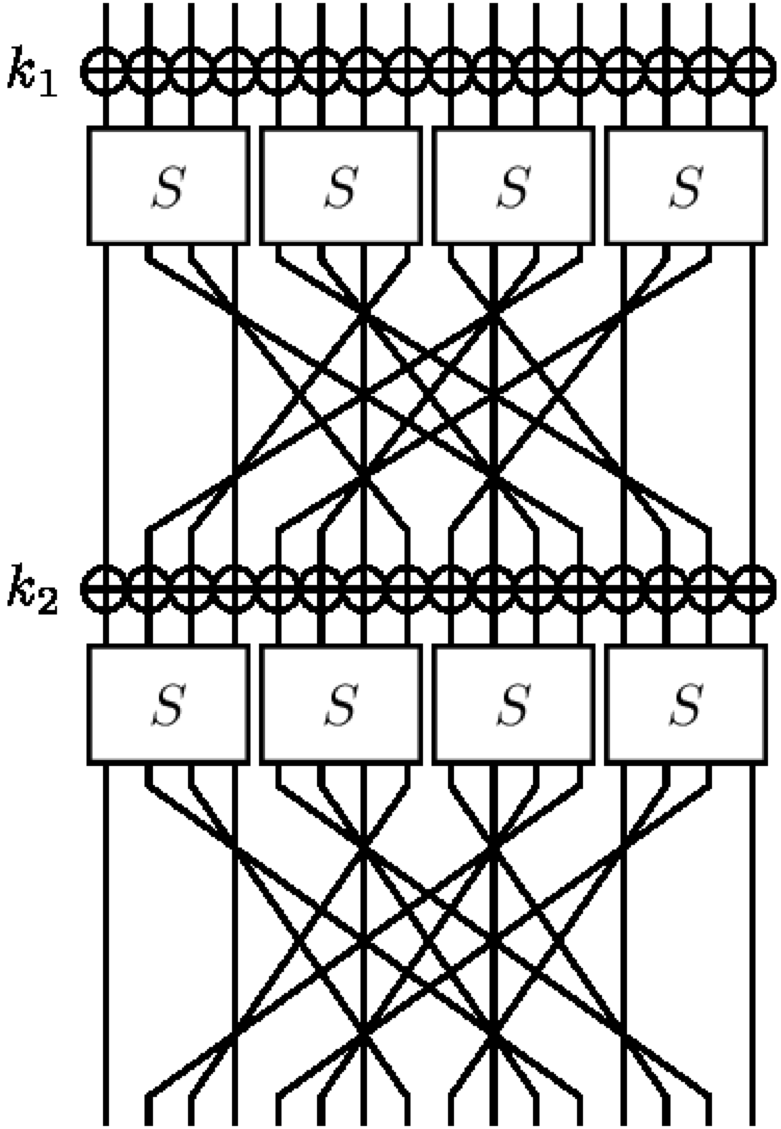
Substitution-permutation networks

S-P network

Gen $S_i : \{0,1\}^{n/l} \Rightarrow \{0,1\}^{n/l}, i = 1, \ldots, l; P : \{0,1\}^n \Rightarrow \{0,1\}^n$, $k_0, k_1, \ldots, k_t$ subkeys

Enc $m_0 = m, c_i = (c_i^1, \ldots, c_i^l) = m_i \oplus k_i, m_{i+1} = P(S_1(c_i^1), S_2(c_i^2), \ldots, S_l(c_i^l))$
$c = m_{t+1}$

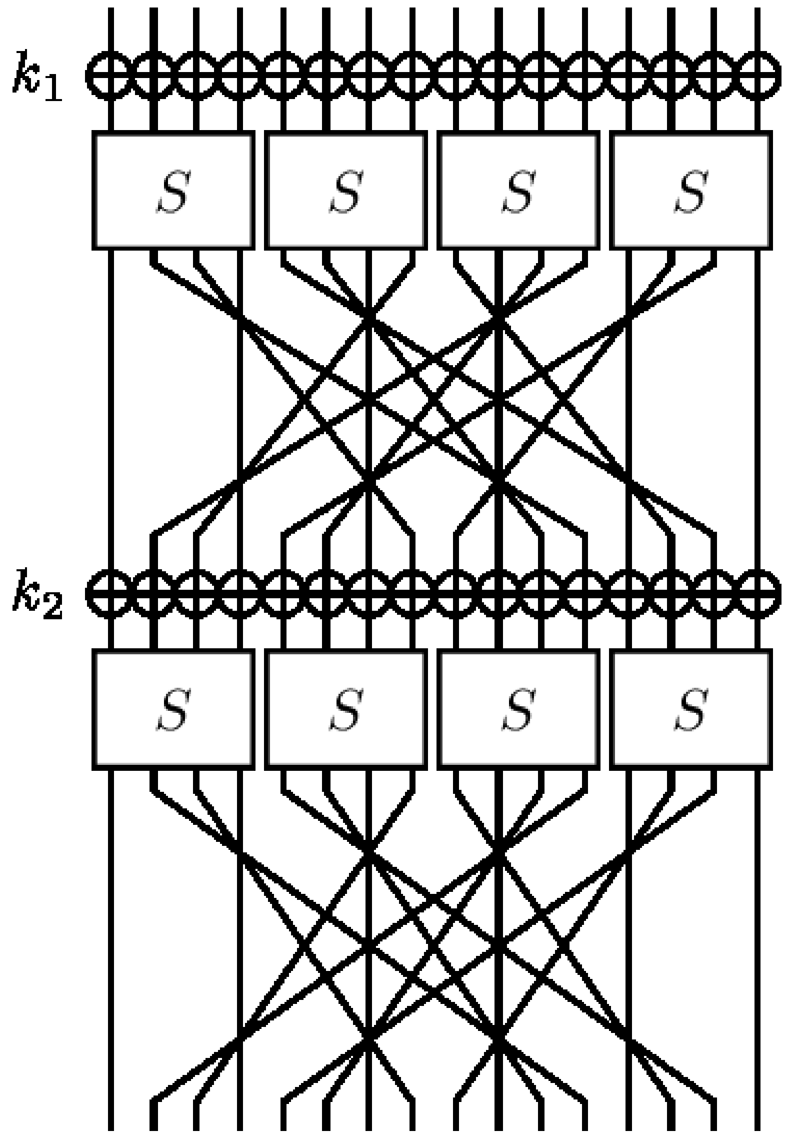
Dec Using $S_i^{-1}, i = 1, \ldots, l, P^{-1}$



Substitution-permutation networks

Properties

- S-box: bijective, avalanche effect, strong output-dependence
- P-box: permutation (shiffling)
- simple, hardware-friendly operations
- Goal: to fulfill the confusion-diffusion paradigm by Shannon
- AES (Rijndael), PRESENT, ...



Feistel vs. S-P

- Feistel: no need for invertible function
- S-P: parallel execution, might be faster on some hardware (not always, e.g. smart cards)
- $\exists$ Feistel combined with S-boxes

Substitution-permutation networks

AES

- Rijndael block cipher's special case (Daemen, Rijmen)
- 2001: USA standard (AES competition)
- software - hardware efficiency
- no practical break known
- best known improvement over brute force: AES-128: reduced to 2^{126} operations
- side-channel attacks (protect hardware!)

Substitution-permutation networks

AES

- block length: 128 bits
- key size: 128 (10 rounds), 192 (12 rounds) or 256 bits (14 rounds)
- 4 x 4 byte-matrices (state)
- KeyExpansion: generate subkeys
- AddRoundKey: ...$\oplus$ subkey
- 9,11 v 13 rounds:
  1. SubBytes (S-box)
  2. ShiftRows (P-box)
  3. MixColumns (P-box)
  4. AddRoundKey
- final round: same without MixColumns

Substitution-permutation networks

AES

- block length: 128 bits
- key size: 128 (10 rounds), 192 (12 rounds) or 256 bits (14 rounds)
- 4 x 4 byte-matrices (state)
- KeyExpansion: generate subkeys
- AddRoundKey: ...$\oplus$ subkey
- 9,11 v 13 rounds:
  1. SubBytes (S-box)
  2. ShiftRows (P-box)
  3. MixColumns (P-box)
  4. AddRoundKey
- final round: same without MixColumns

AES background (theory)

Rijndael-test

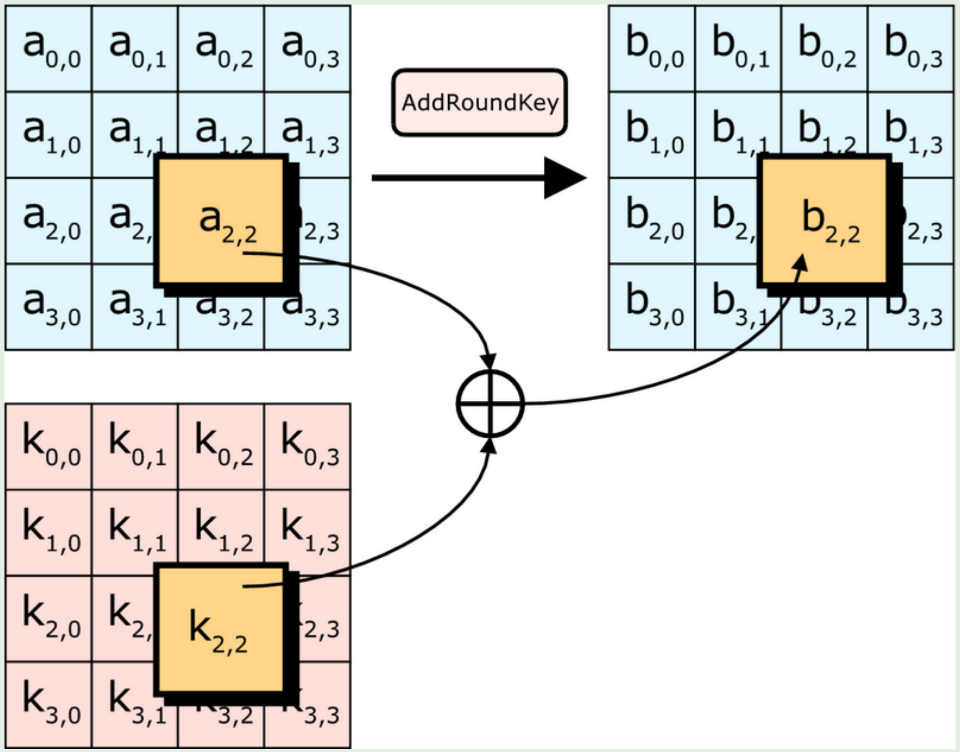
- $x^8 + x^4 + x^3 + x + 1$ polynomial over $\mathbb{Z}_2$ (irreducible)
- $\mathbb{F} = \mathbb{Z}_2[x]/\langle x^8 + x^4 + x^3 + x + 1 \rangle$: 256-element field (4 basic operations on bytes)
- string to polynomial:
  $b = (b_0, b_1, \ldots, b_7) \longleftrightarrow b_0 + b_1 x + \cdots + b_7 x^7 \in \mathbb{F}$

AES key generation and expansion

KeyExpansion

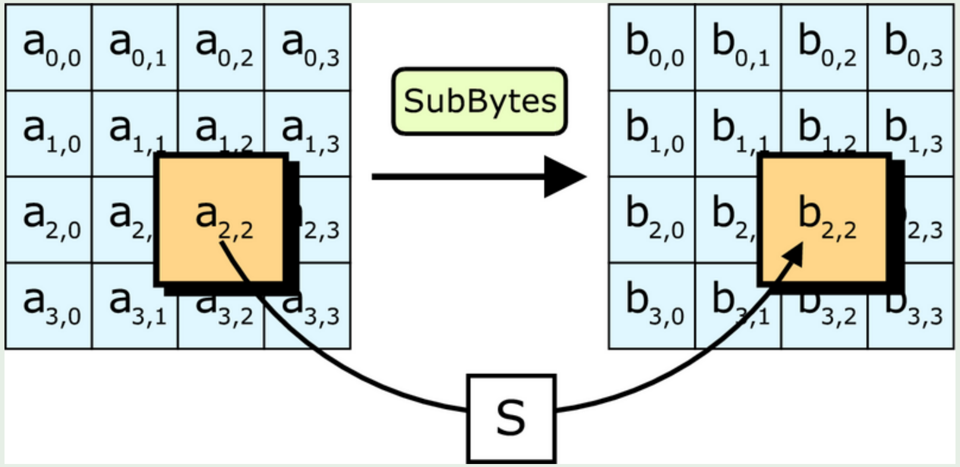
- subkeys for all rounds
- $k_{0,0}, \ldots, k_{0,3}, k_{1,0}, \ldots, k_{3,3}$
- skipping details

AddRoundKey



AES S-box

SubBytes

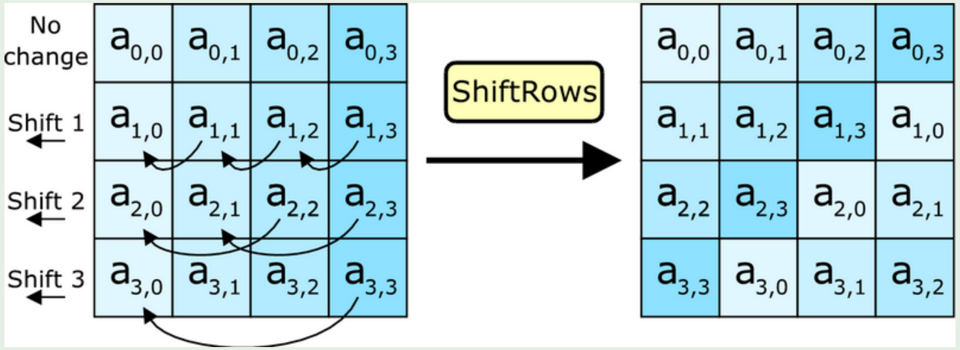


SubBytes

- $x \in \{0,1\}^8 : S(x) = Cx^{-1} + c : x^{-1} \in \mathbb{F}$
- $C \in \{0,1\}^{8x8}$ fixed invertible matrix
- $c \in \{0,1\}^8$ fixed vector
- $S(.)$ affine transformation without fixed point (linear + constant)
- inverse: $x = S^{-1}(y) = (C^{-1}(y-c))^{-1}$

AES P-box (1st half)

ShiftRows

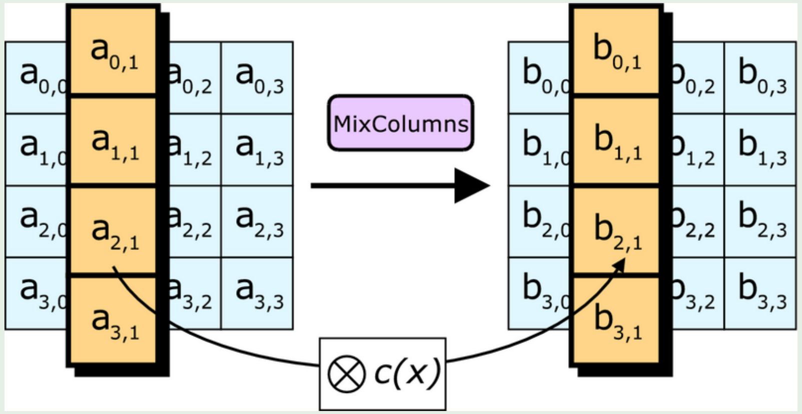


ShiftRows

- $<<< i$ in row i
- inverse easy

AES P-box (2nd half)

MixColumns



ShiftRows

- $a(x) = a_{3,1}x^3 + a_{2,1}x^2 + a_{1,1}x + a_{0,1}$
- $c(x) = 3x^3 + x^2 + x + 2$
- $a \otimes c \equiv a \cdot b \mod (x^4 + 1)$
- inverse: $c^{-1}(x) = 11x^3 + 13x^2 + 9x + 14 \mod (x^4 + 1)$