Block Ciphers and meet-in-the-middle attack.

1. substitution cipher

a substitution on alphabet A

Latin $|A| = 26$

frequency analysis

language redundancy.

2. Vigenère cipher

a substitution on alphabet A^t

26^t

pl.-text block $m_1\ m_2 \dots m_t \in A^t$

cipher-text block $P_1(m_1)\ P_2(m_2) \dots P_t(m_t)$

does not provide diffusion within the block.

$P_i(m_i)$ does not depend on m_j, $j \neq i$.

cryptanalysis is reduced to frequency analysis on A.

Modern Block Ciphers

1) combinations of several rounds (transforms)

2) each round constructed with

substitutions, permutations (transpositions), linear transforms (linear layers).

Notions

1) N number of rounds

2) key schedule

K cipher key, binary string.

$K_1, K_2, \ldots, K_N$

K_i is used in i-th round.
constructed from K.
by a public algorithm.

3) round function

input S_{r-1} current state
K_r r-th sub key.

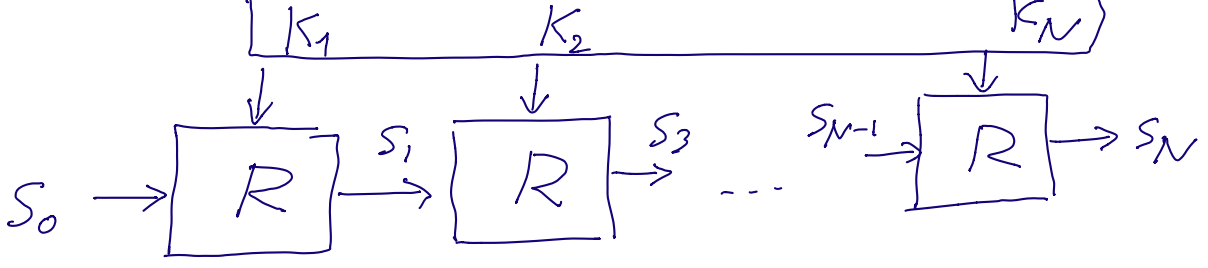
output S_r next state.

$$S_r = R(S_{r-1}, K_r)$$

4) block size all states
$S_0, \ldots, S_{r-1}, \ldots, S_N$
are bit strings of length n.

Encryption

initial state $S_0 = x$ pl-text n-bit block.
final state $S_N = y$ cipher-text n-bit block.

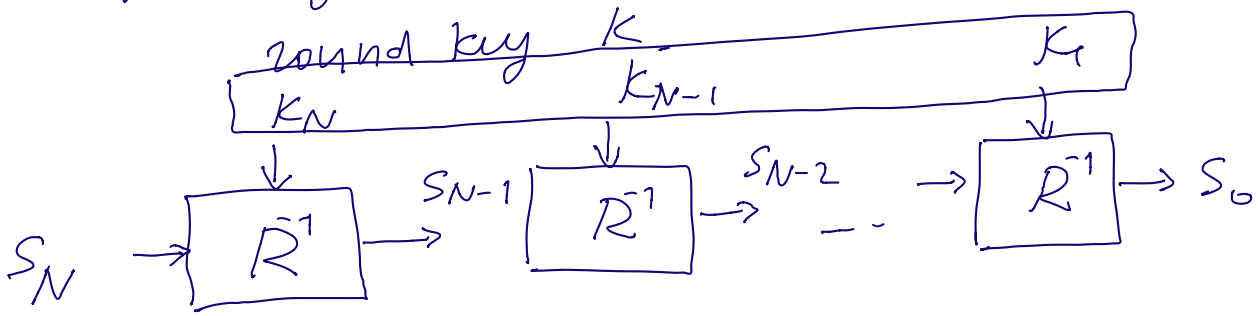


in practice, round functions may differ in different rounds.

Decryption

property $R^{-1}(R(S, K), K) = S$

for any state S



because $S_{N-1} = R^{-1}(S_N, K_N) =$

$= R^{-1}(R(S_{N-1}, K_N), K_N) =$

$= S_{N-1}$

Convenient to have $R = R^{-1}$

enc. algorithm ≡ dec. algorithm.

in DES (digital encryption standard)

not in AES

Two main constructions of block ciphers:

1) Feistel cipher like DES
2) Substitution Permutation Network (SPN) AES.

DES

$N = 16$ number of rounds

key size 56

blocksize $n = 64$

key schedule $K_1\ K_2\ \dots\ K_{16}$

48-bit selections from 56-bit main cipher key.

initial state $S_0 = (\underbrace{X_0}_{32}, \underbrace{X_1}_{32\text{-bit}})$ ← 64-bit pl-text block.

internal states $S_r = (X_r, X_{r+1})$ $r = 1, \dots, 15$

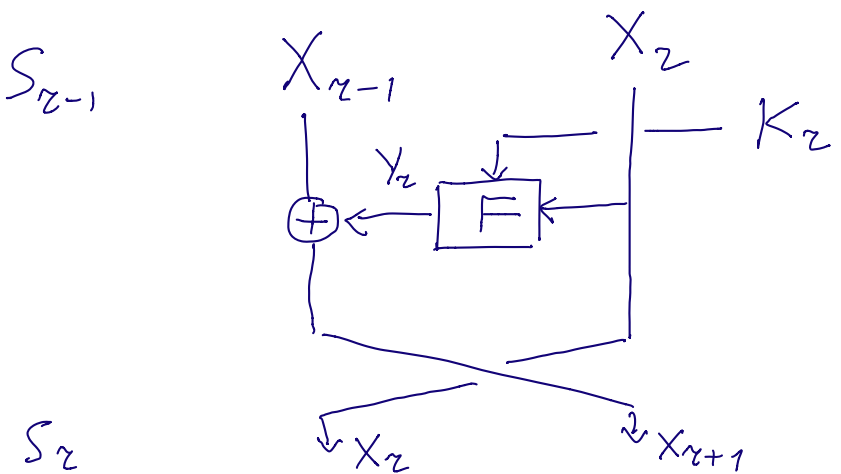
final state $S_{16} = (X_{17}, X_{16})$

↑ cipher-text block

Rounds in DES

15 regular rounds

1 irregular round (last round)



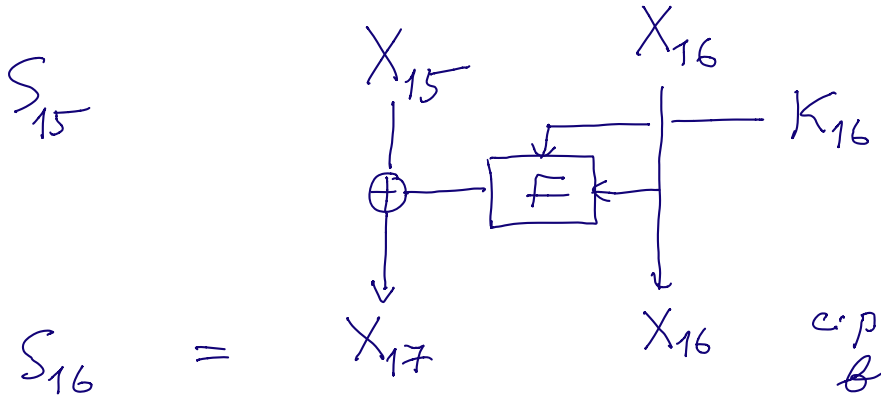
$r = 1, \dots, 15$

$Y_2 = F(X_2, K_2)$ constructed with 8 DES S-boxes

(X_2: 32-bit, K_2: 48-bit)

each S-box : 6-bit $\longrightarrow$ 4-bit.

last round



Substitution-Permutation Network (SPN)

ℓ, m, N naturals

$n = \ell \cdot m$ block size

$N+1$ number of rounds

$N-1$ regular rounds

2 irregular last rounds.

SPN S-box : ℓ-bit $\longrightarrow$ ℓ-bit

invertible

P permutation on $1, 2, \ldots, n$

states in SPN

$X_0, X_1, \dots, X_{N+1}$ n-bit strings

X_0 ↑ pl-text block

X_{N+1} ↑ cipher-text block.

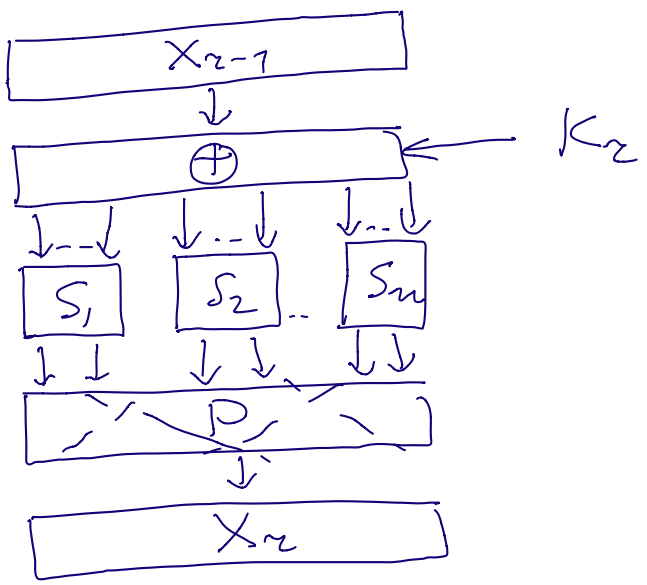
regular round in SPN

$$X_r = P\left[(S_1 \dots S_m)(X_{r-1} \oplus K_r)\right]$$

↑ S-boxes $S_1 = \dots = S_m = S$

$r = 1, \dots, N-1$

by diagram



irregular rounds

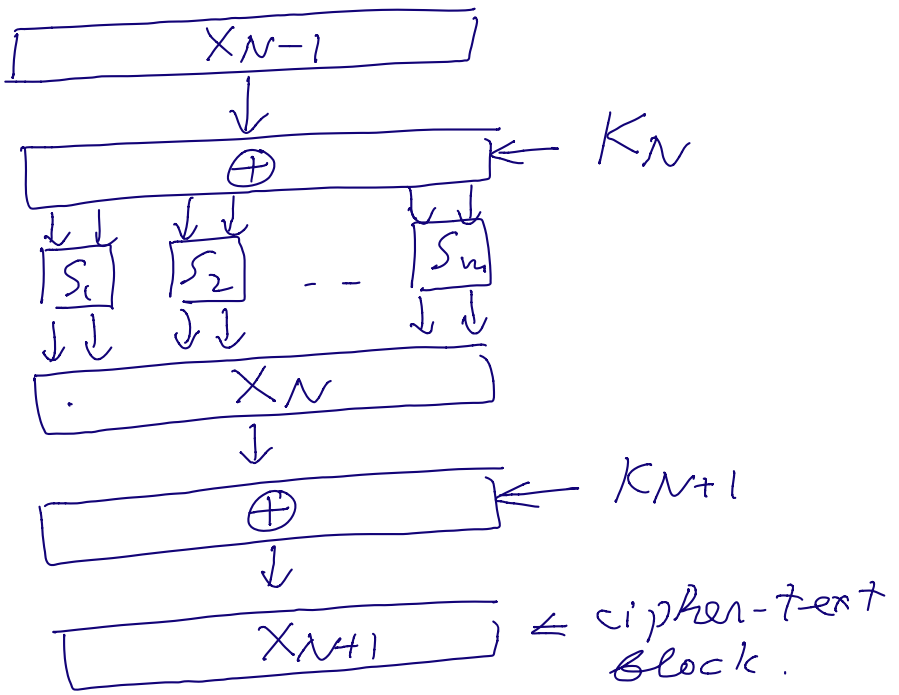
1) N-th round

$$X_N = (S_1 \dots S_m)(X_{N-1} \oplus K_N)$$

2) N+1-th round

$$X_{N+1} = X_N \oplus K_{N+1}$$

by diagram



Why do we have irregular rounds in SPN?

P, $S_1, \ldots, S_m$ public

P in round N

P, $S_1 \ldots S_m$ in round N+1

do not add security to the transform.

Meet-in-the-Middle Attack.

known pl.-text attack

given pl.-text/c.-text pairs

X_1, Y_1
X_t, Y_t

recover cipher key K, $|K| = u$.

brute force takes 2^u trials.

find K faster than that?

works when

1) round keys K_i are bit selections of bits K.
2) this selection is not proper.

choose $1 \le i < N$ (# rounds)

X - pl.-text block
Y c.-text block

$$\downarrow \begin{cases} S_0 = X \\ S_1 = R(S_0, K_1) \\ \vdots \\ S_i = R(S_{i-1}, K_i) \end{cases}$$

$$\uparrow \begin{cases} S_i = R^{-1}(S_{i+1}, K_{i+1}) \\ \vdots \\ S_{N-1} = R^{-1}(S_N, K_N) \\ S_N = Y \end{cases}$$

S_i may be computed from X and from Y

subkeys: $K(1) = K_1 \cup \dots \cup K_i$
all key-bits in $K_1, \dots, K_i$

$K(2) = K_{i+1} \cup \dots \cup K_N$
all key-bits in $K_{i+1} \dots K_N$

$K(1) \cap K(2)$ both
all key-bits in $K(1)$ and $K(2)$

$|K(1)| = u_1, \quad |K(2)| = u_2, \quad |K(1) \cap K(2)| = u_{12}$

$S_i = S_i(X, K(1))$
$S_i = S_i(Y, K(2))$

Algorithm to find K.

1. For each u_1-bit $K(1)$ compute
$S = S_i(X, K(1))$ in n
keep pair $(\overline{K(1)}, \overline{S})$
(2^{u_1} such pairs)

2. sort the pairs by
$(\underbrace{K(1) \cap K(2)}_{u_{12}}, \underbrace{S}_{n})$

3. For u_2-bit $K(2)$ compute
$S = S_i(Y, K(2))$

look up $(K(1) \cap K(2), S)$ in sorted pairs

3.1. match compute $K = K(1) \cup K(2)$ candidate for cipher key

3.2. no match, repeat 3.

Complexity 1. 2^{u_1}

2. $u_1 \cdot 2^{u_1}$

3. 2^{u_2}

overall is $2^{u_1} + u_1 \cdot 2^{u_1} + 2^{u_2} < 2^{u}$

if $\boxed{u_1, u_2 < u}$. advantage over brute force.

Remarks on implementation.

1. $u = |K| > n$ (block size)

(key size) $u - n$

candidates is ≈ 2

1.1 are brute forced.

1.2 apply algorithm to $x \leftarrow (x_1, x_2)$

$Y \leftarrow (Y_1, Y_2)$
instead of $\overset{x}{\underset{y}{}}$

2. Algorithm variation

pairs $\underline{(K(1), S)}$, $S = S_i(X, K(2))$

$(K(2), S)$ $S = S_i(Y, K(2))$

sort all the pairs
by substring $(K(1) \cap K(2), S)$

find matches.

one match (the same $(K(1) \cap K(2), S)$)
given one candidate for
the cipher key.

as effective as the previous
method by requires more
memory.

In the assignment

SPN with parameters

$l = m = N = 4$,

Block size $n = l \cdot m = 16$

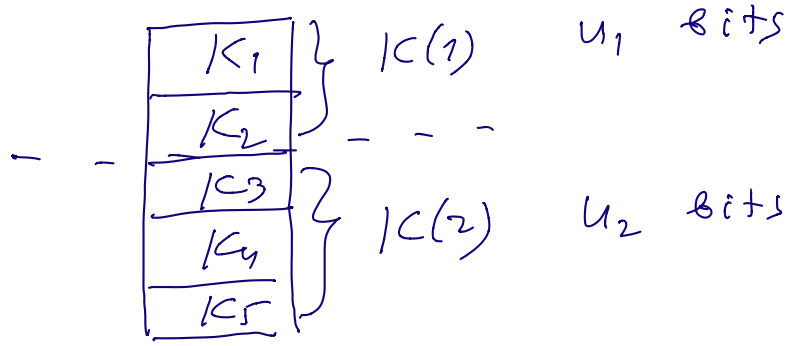
SPN with $N+1=5$ rounds

key size $=32$ bits

given 4 pl-text/c-text pairs

find 32-bit cipher key.

split encryption into 2 parts



s.t. $u_1 \approx u_2 < 32$

if use only 1 pl-text/c-text pairs

candidate solutions $\approx 2^{32-16} = 2^{16}$

2 pl-text/c-text pairs

candidate solutions $\approx 2^{32-32} = 1$.