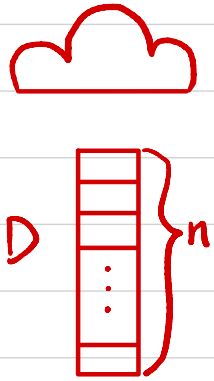
CSCI 1515 Applied Cryptography

This Lecture:

- Private Information Retrieval (Continued)
- Bootstrapping SWHE to FHE
- Practical Constructions of Block Cipher

Private Information Retrieval (PIR)



WANT: D[i]

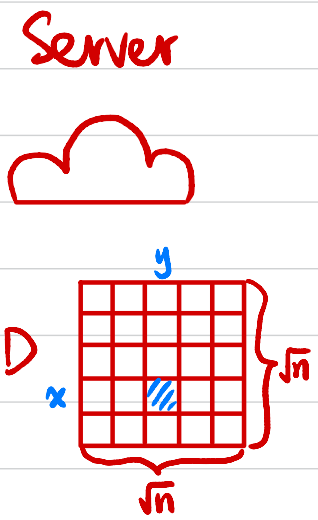
While hiding i against Server

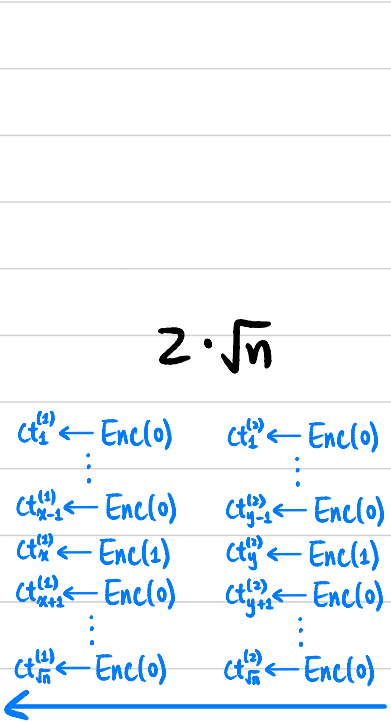
Trivial Solution:

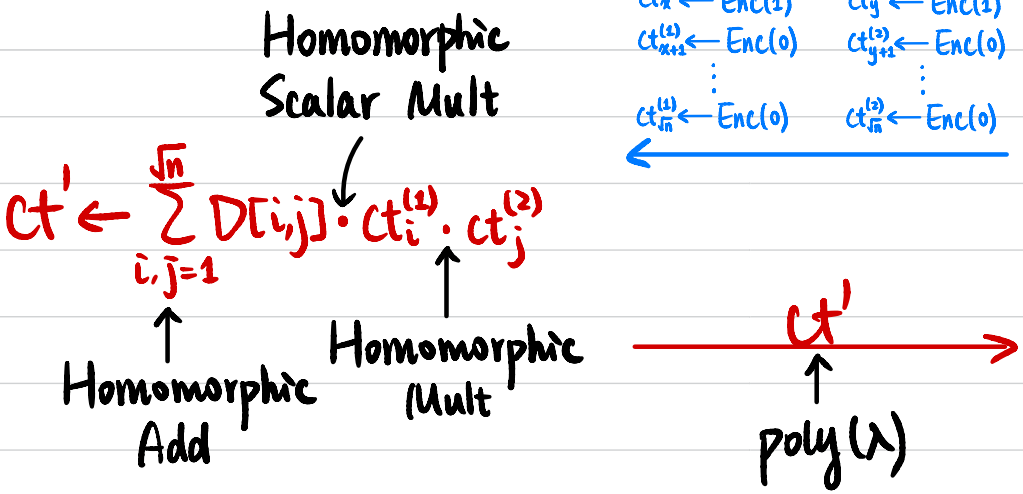
Communication complexity $O(n)$

Goal: Communication complexity $o(n)$

Private Information Retrieval (PIR)







D[x,y] = Dec(ct')

Extend to dimension d ? Communication $d \cdot \sqrt[d]{n}$

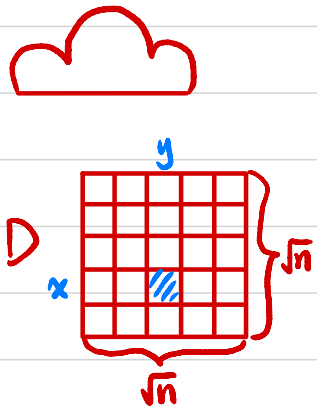
Homomorphic Mult = (d-1)·n

Homomorphic Scalar Mult = n

Homomorphic Add = n

PIR from Additive HE

Server



Client

$ct_1 \leftarrow Enc(0)$

$\vdots$

$ct_{x-1} \leftarrow Enc(0)$

$ct_x \leftarrow Enc(1)$

$ct_{x+1} \leftarrow Enc(0)$

$\vdots$

$ct_{\sqrt{n}} \leftarrow Enc(0)$

WANT: $D[x,y]$

While hiding (x,y) against Server

Homomorphic Scalar Mult

$$\forall j \in [\sqrt{n}],\ ct'_j \leftarrow \sum_{i=1}^{\sqrt{n}} D[i,j] \cdot ct_i$$

Homomorphic Add

$ct'_1, \cdots, ct'_{\sqrt{n}}$

$D[x,y] = ?$

Application: Secure 2PC ?

Alice

Bob

$C(x, y)$

Input: x

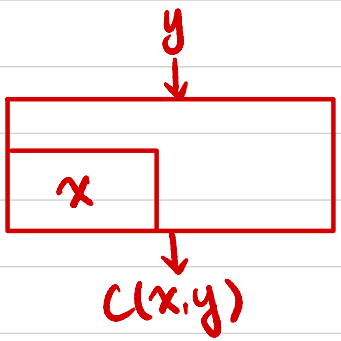
Input: y

$ct \leftarrow Enc(y)$

$ct' \leftarrow Eval(f, ct)$

$f_x(y) = C(x, y)$

$f(y) \leftarrow Dec_{sk}(ct')$



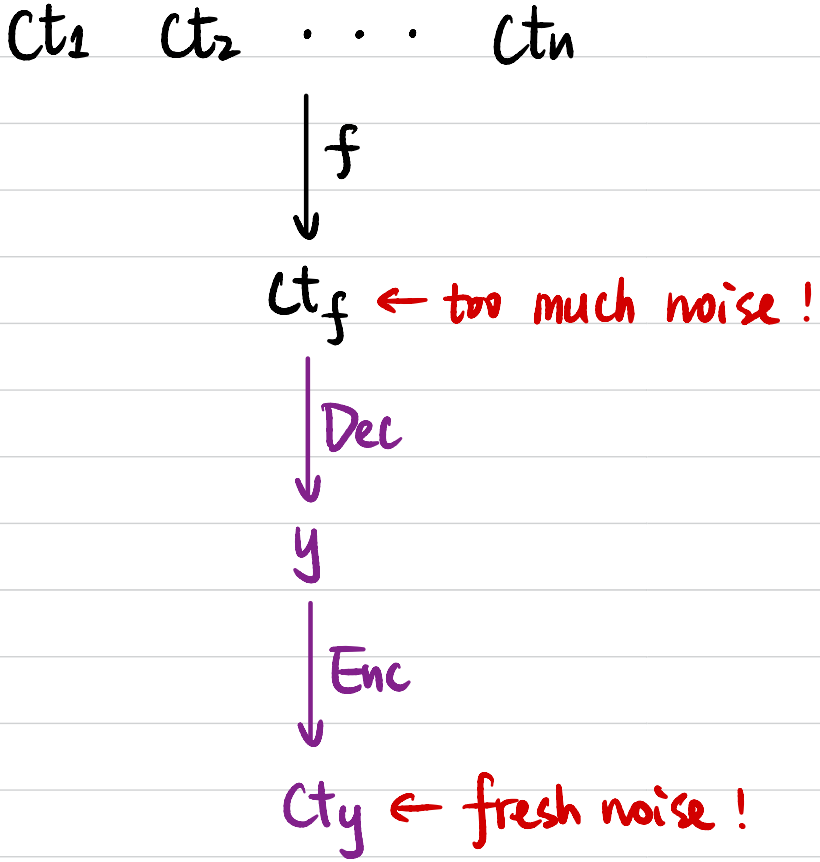
FHE Constructions

Step 1: Somewhat Homomorphic Encryption (SWHE)

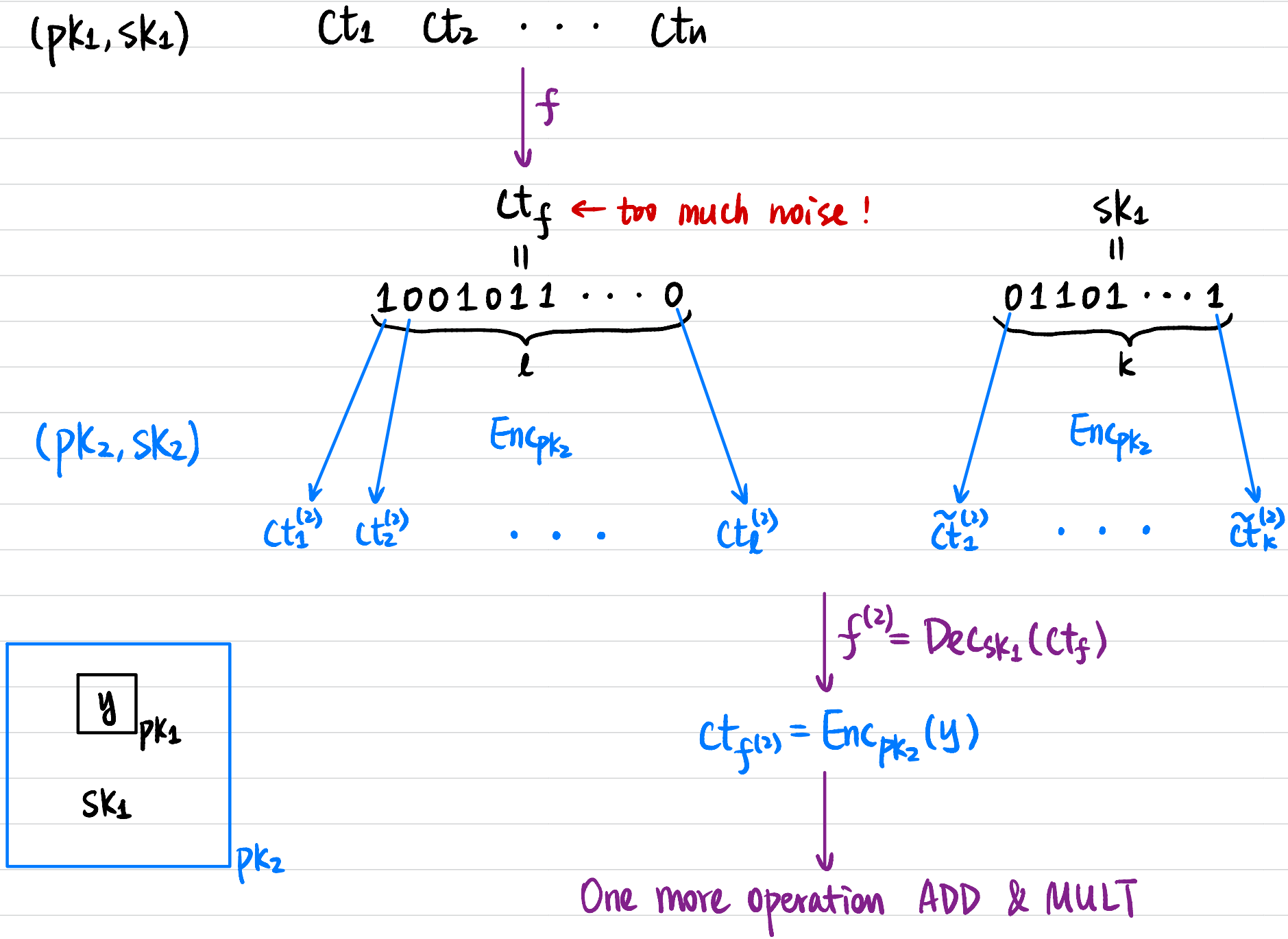
- over Integers
- from LWE (GSW)
- from RLWE (BFV)

Step 2: Bootstrapping

Step 2: Bootstrapping



Leveled FHE


(pk_1, sk_1)
Ct_1 Ct_2 $\cdots$ Ct_n
f
Ct_f ← too much noise !
=
1001011 $\cdots$ 0
ℓ
sk_1
=
01101 $\cdots$ 1
k
(pk_2, sk_2)
Enc_{pk_2}
Enc_{pk_2}
$Ct_1^{(2)}$ $Ct_2^{(2)}$ $\cdots$ $Ct_\ell^{(2)}$
$\tilde{Ct}_1^{(2)}$ $\cdots$ $\tilde{Ct}_k^{(2)}$
$f^{(2)} = Dec_{sk_1}(Ct_f)$
$Ct_{f^{(2)}} = Enc_{pk_2}(y)$
One more operation ADD & MULT
y
pk_1
sk_1
pk_2

Step 2: Bootstrapping

Leveled FHE: pk_1, pk_2, pk_3, . . . , pk_n

$Enc_{pk_2}(sk_1)$ $Enc_{pk_3}(sk_2)$ $Enc_{pk_n}(sk_{n-1})$

FHE: pk. $Enc_{pk}(sk)$

"circular secure" assumption

Block Cipher

$$F: \{0,1\}^{\lambda} \times \{0,1\}^{n} \to \{0,1\}^{n}$$

λ: key length

n: block length

It is assumed to be a pseudorandom permutation (PRP).

Construction: Advanced Encryption Standard (AES)

- $\lambda = 128/192/256$, $n = 128$
- Standardized by NIST in 2001
- Competition 1997-2000

Before AES: Data Encryption Standard (DES)

- $\lambda = 56$, $n = 64$

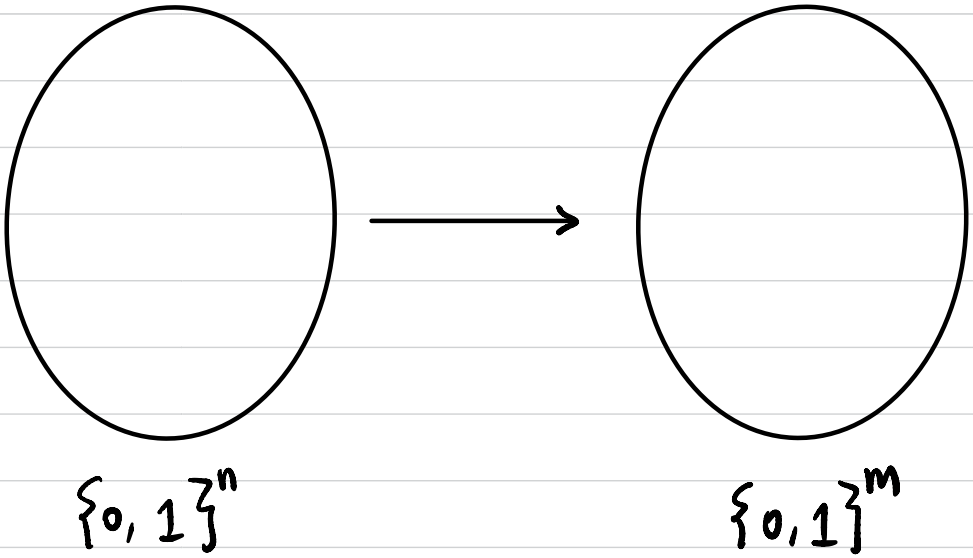
Pseudorandom Function (PRF)

Keyed Function $F: \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^m$

$F(k, x) \to y$

- F: deterministic poly-time
- k: key
- x: input
- y: output

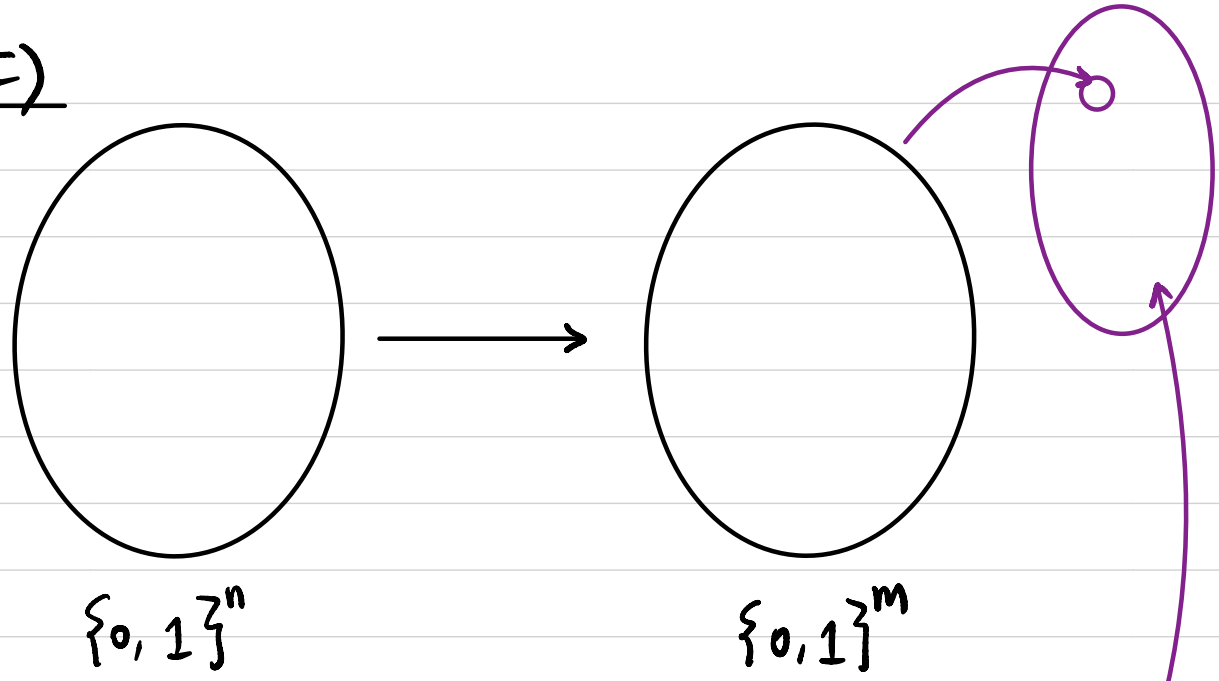
$k \xleftarrow{\$} \{0,1\}^\lambda$ $\quad F_k:$



"looks like a random function"

Pseudorandom Function (PRF)

$k \xleftarrow{\$} \{0,1\}^\lambda$ $\quad F_k:$



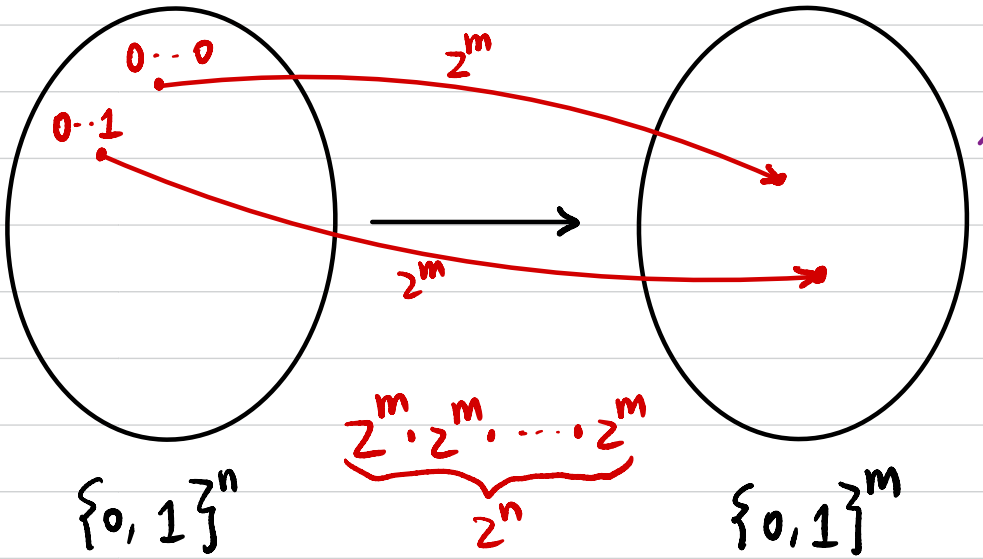
How many possible F_k's ?

2^λ

$\{\{C$ (not knowing k)

$f \xleftarrow{\$} \{ F \mid F: \{0,1\}^n \to \{0,1\}^m \}$

$f:$

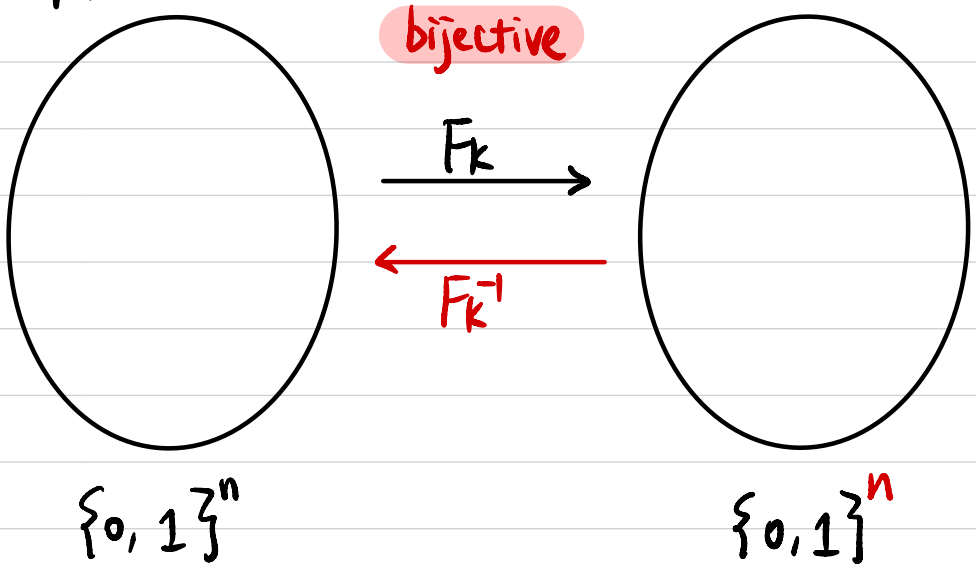


How many possible f's ?

$(2^m)^{2^n}$

Pseudorandom Permutation (PRP)

$k \xleftarrow{\$} \{0,1\}^\lambda$ $\quad F_k :$



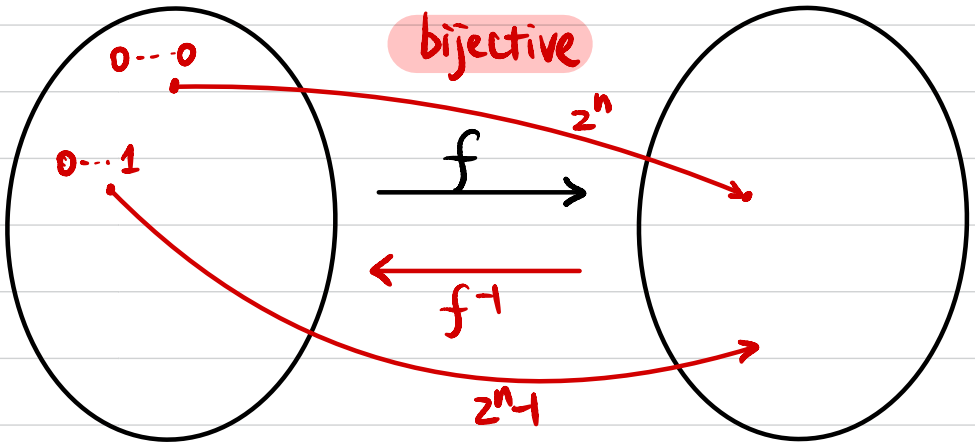
How many possible F_k's ?

2^λ

$\{\{C$ (not knowing k)

$f \xleftarrow{\$} \{ F \mid F: \{0,1\}^n \to \{0,1\}^n,$ F is bijective $\}$

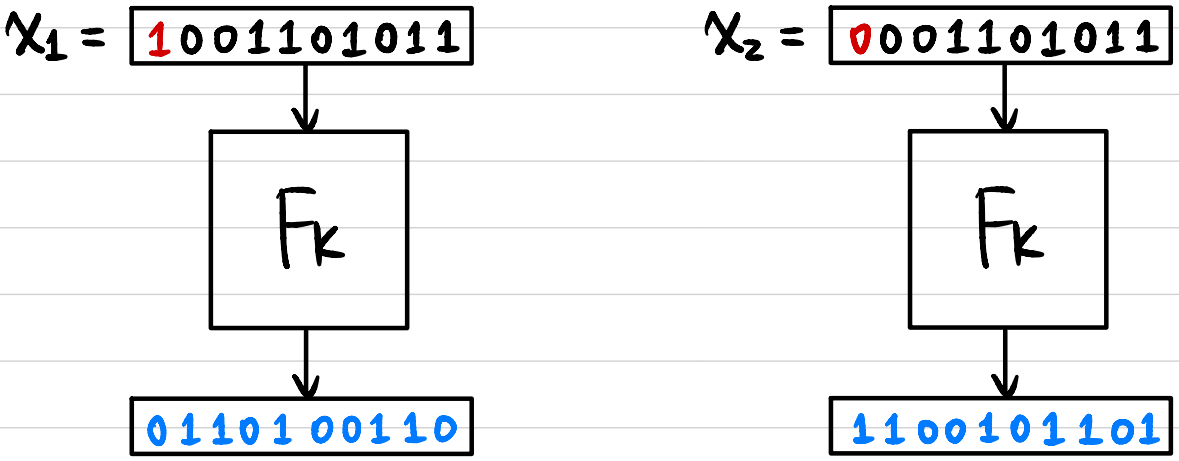
$f :$



How many possible f's ?

$2^n!$

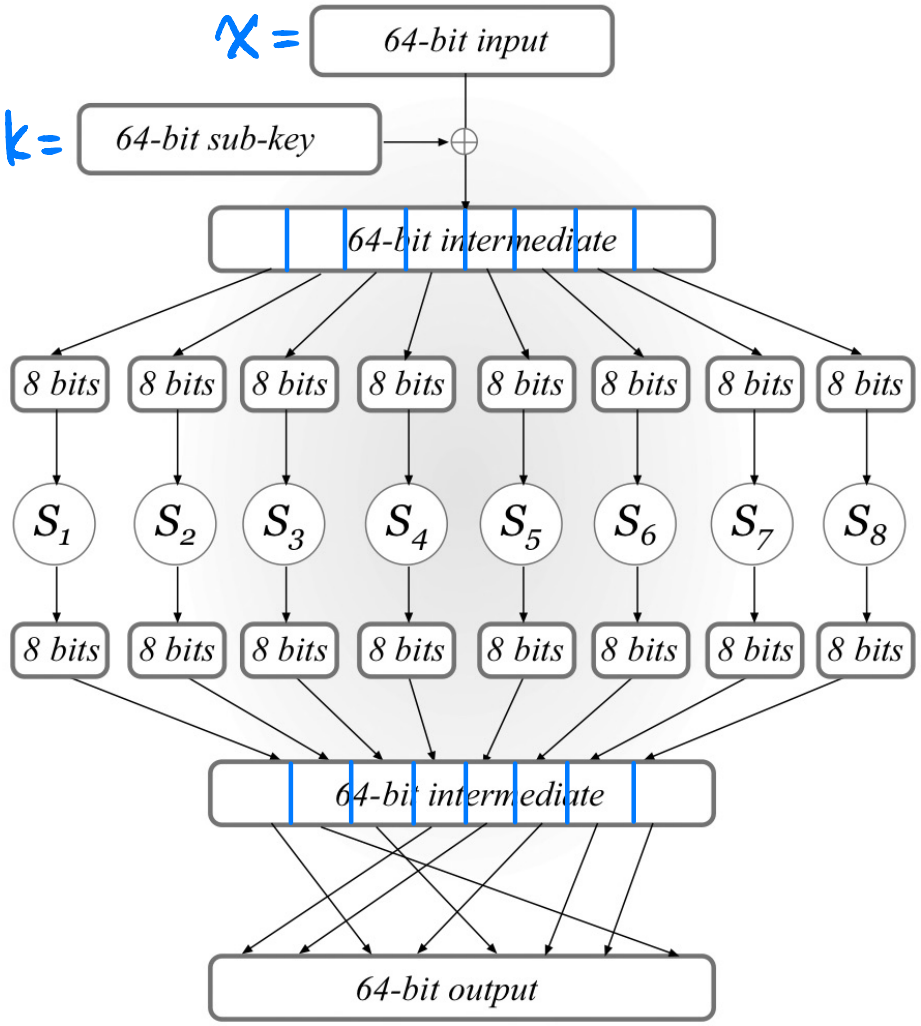
Substitution-Permutation Network (SPN)



Design Principle: "Avalanche Effect"

A one-bit change in the input should "affect" every bit of the output.

Substitution-Permutation Network (SPN)



A single round of SPN

"Confusion-Diffusion Paradigm"

Step 1: Key Mixing

$x := x \oplus k$

Step 2: Substitution (Confusion Step)

$S_i : \{0,1\}^8 \rightarrow \{0,1\}^8$ (S-box)

Public permutation / one-to-one map

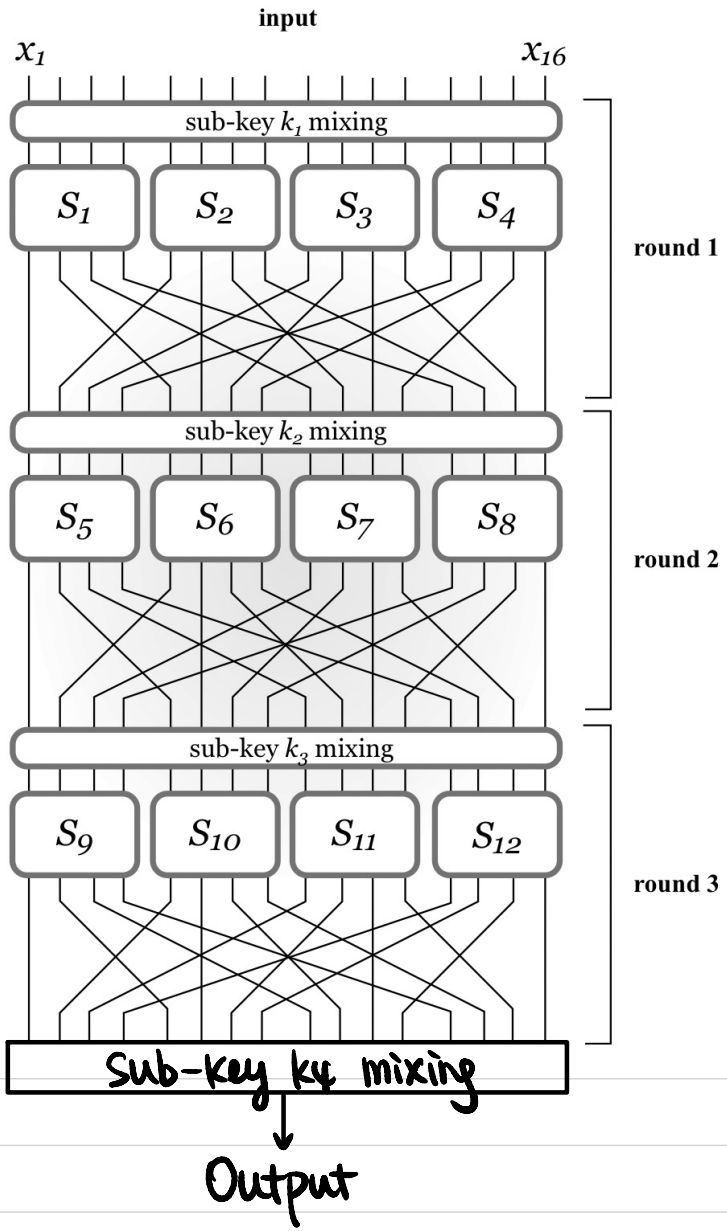
1-bit change of input
→ at least 2-bit change of output

Step 3: Permutation (Diffusion Step)

$P : [64] \rightarrow [64]$

Public mixing permutation
↓
affect input to multiple S-boxes next round

Substitution-Permutation Network (SPN)



3-round SPN:

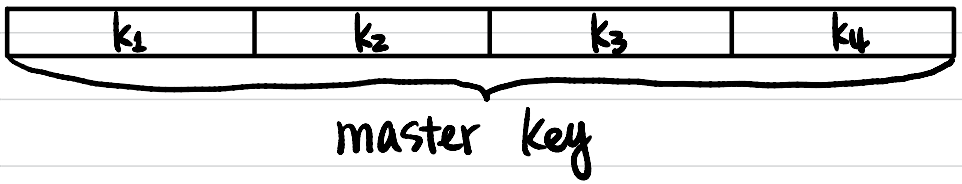
3-round
- key mixing
- substitution
- permutation

1 final-round key mixing

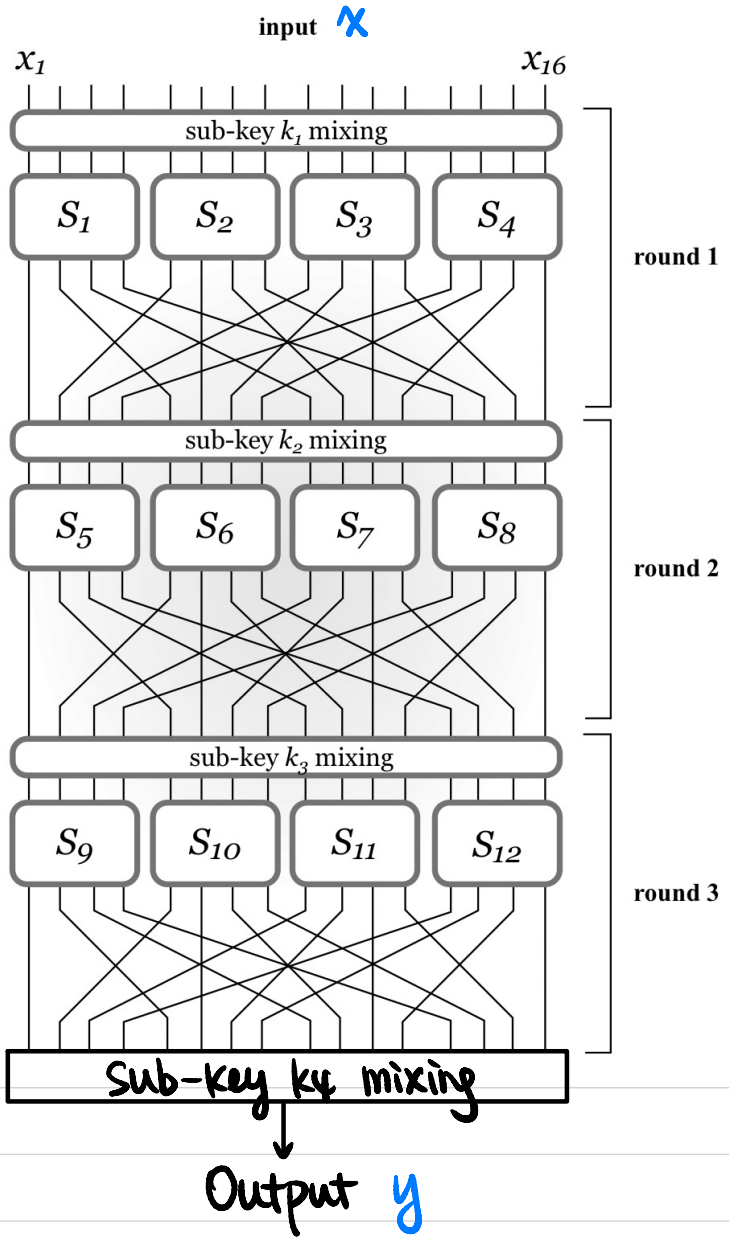
Key Schedule:

How we derive sub-keys from master key.

Example:



Substitution-Permutation Network (SPN)



An SPN is invertible given the master key.

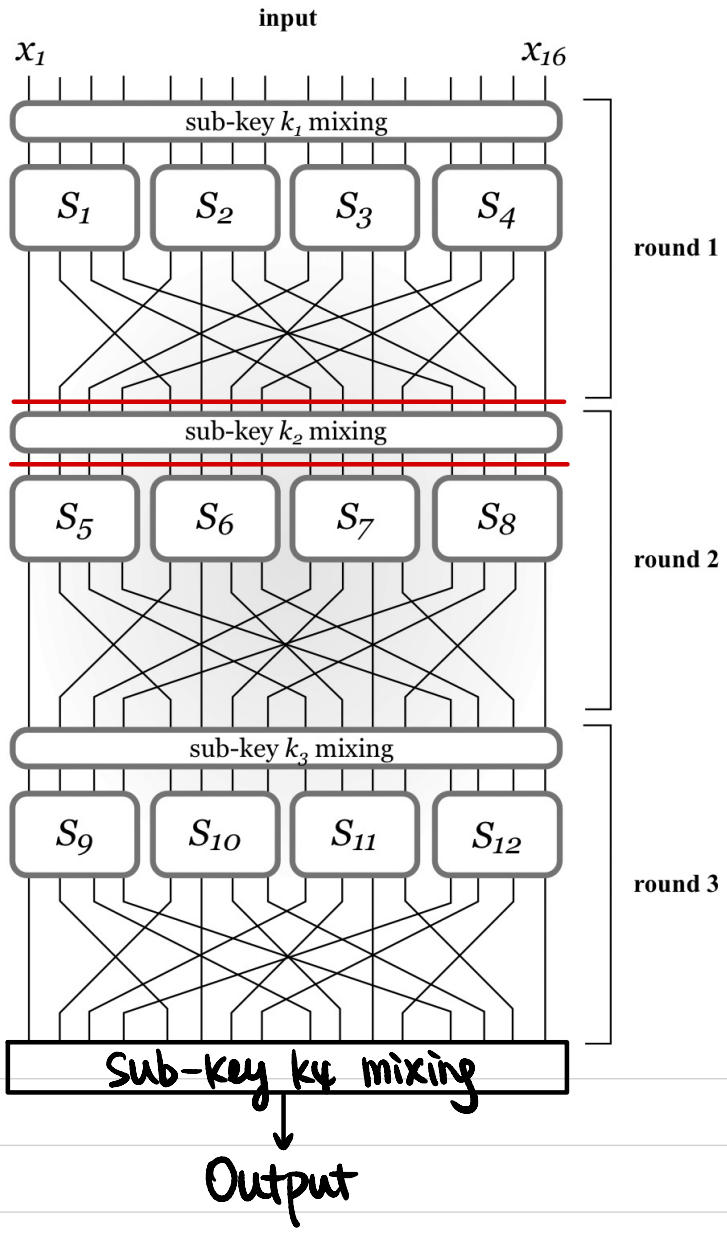
⇓

permutation

How to compute $F_k^{-1}(y)$?

Why do we need a final key mixing step ?

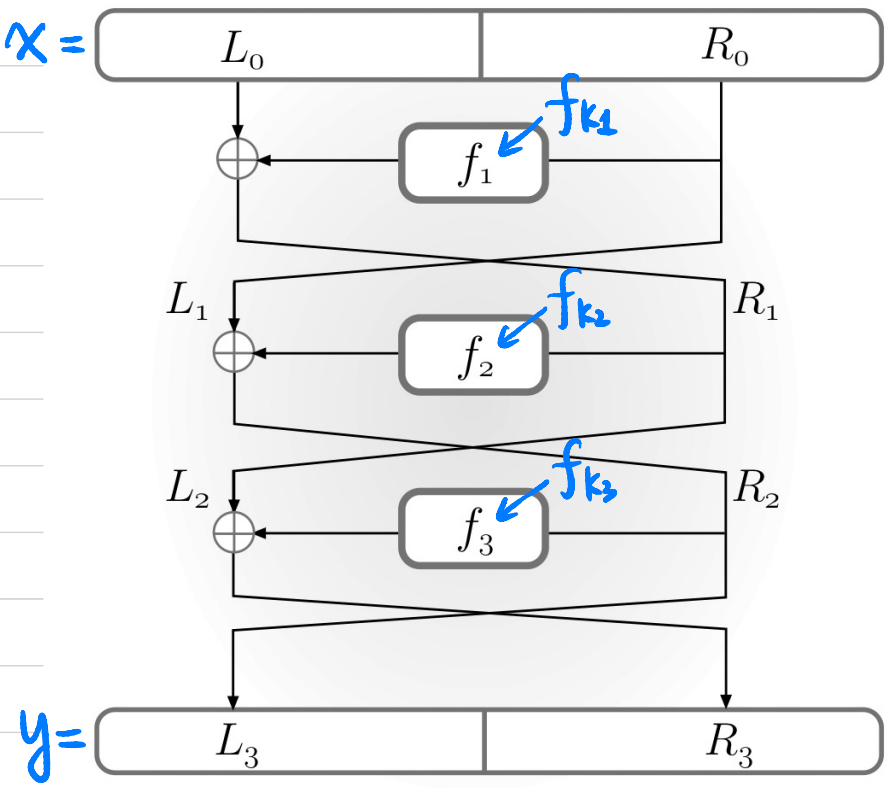
Attacks on Reduced-Round SPN



1-round SPN w/o final key mixing ?

1-round SPN w/ final key mixing ?

Feistel Network



3-round Feistel Network

$f_{k_i}: \{0,1\}^{n/2} \to \{0,1\}^{n/2}$

↑ round function

How to compute $F_k^{-1}(y)$?

Attacks on reduced-round Feistel Network

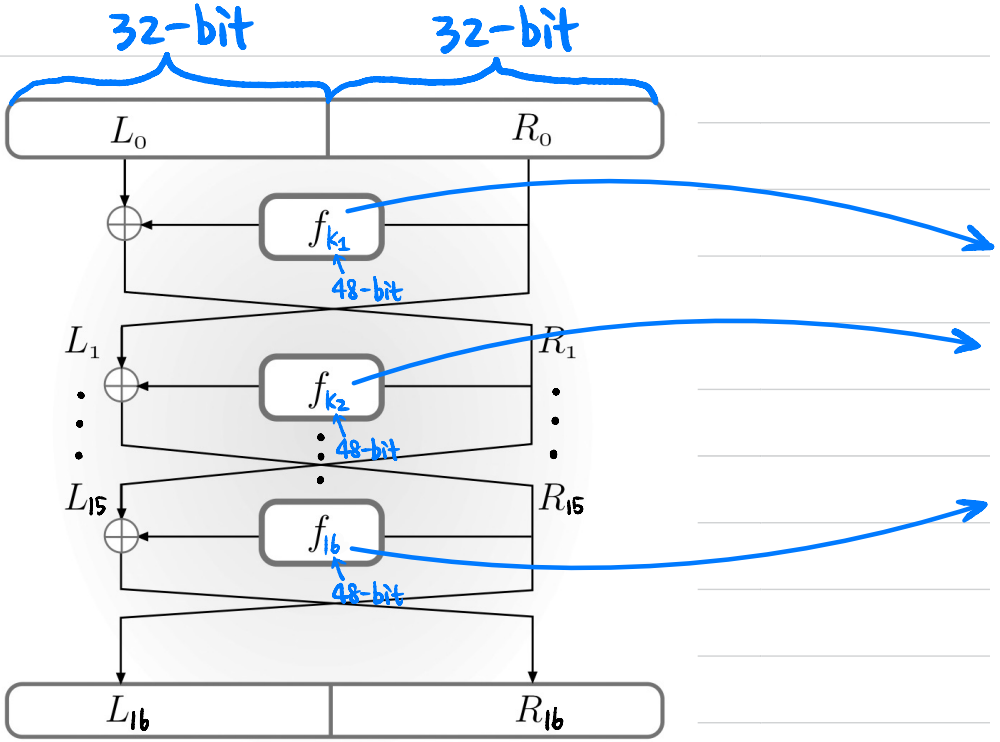
Data Encryption Standard (DES)

$F: \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^n$

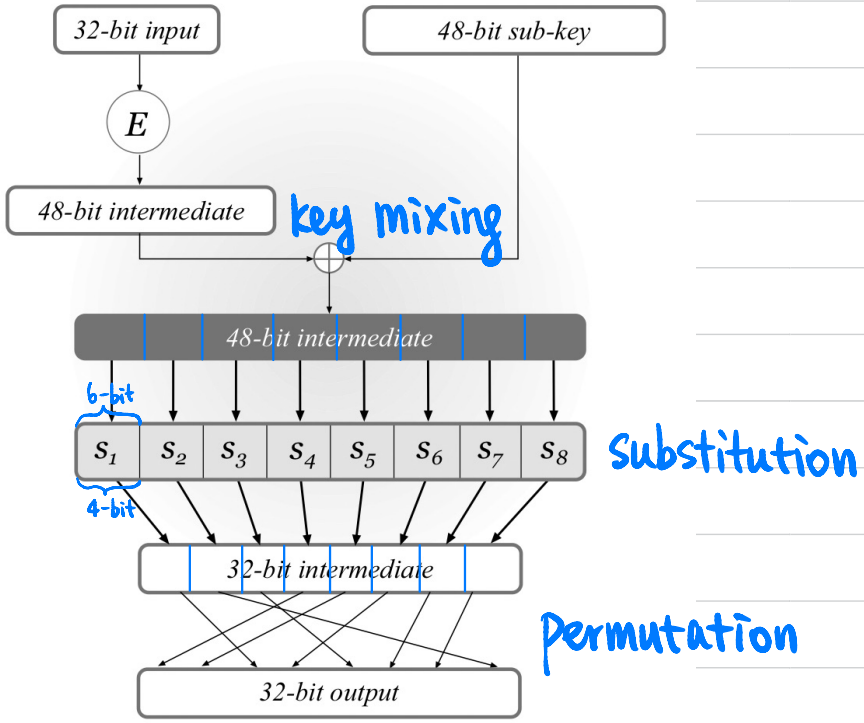
block length $n=64$

master key length $\lambda=56$

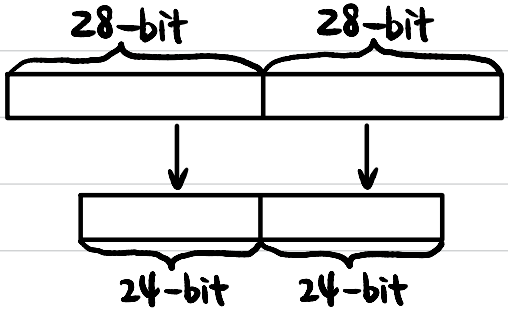
16-round Feistel Network



DES mangler function

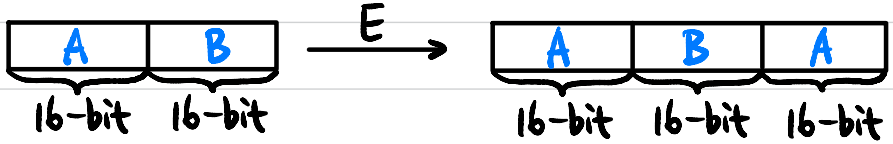


Key Schedule:



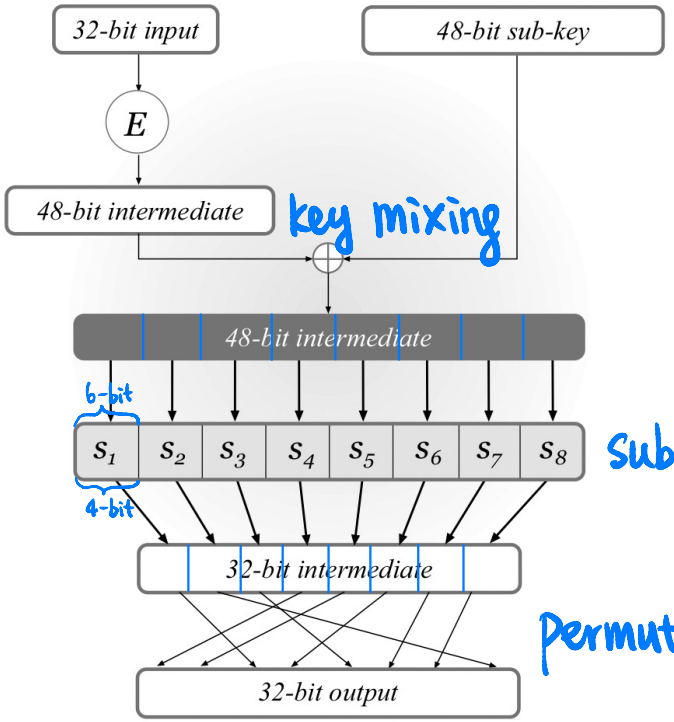
(master key)

E: expansion function



Data Encryption Standard (DES)

DES mangler function



S-box: $\{0,1\}^6 \rightarrow \{0,1\}^4$

① "4-to-1":

Exactly 4 inputs map to same output

② 1-bit change of input

→ at least 2-bit change of output

Mixing Permutation: $[32] \rightarrow [32]$

4 bits from each S-box will affect the input to 6 S-boxes in the next round