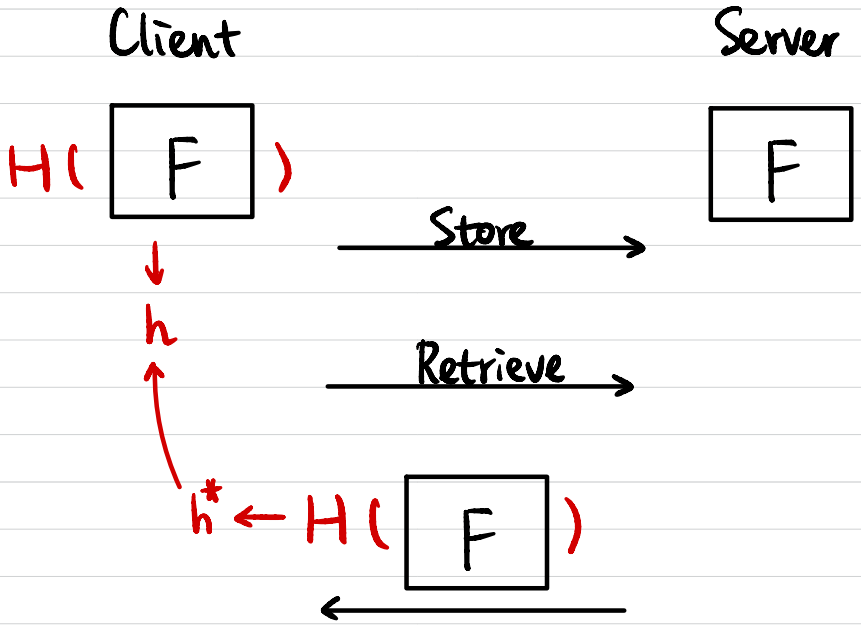
CSCI 1510

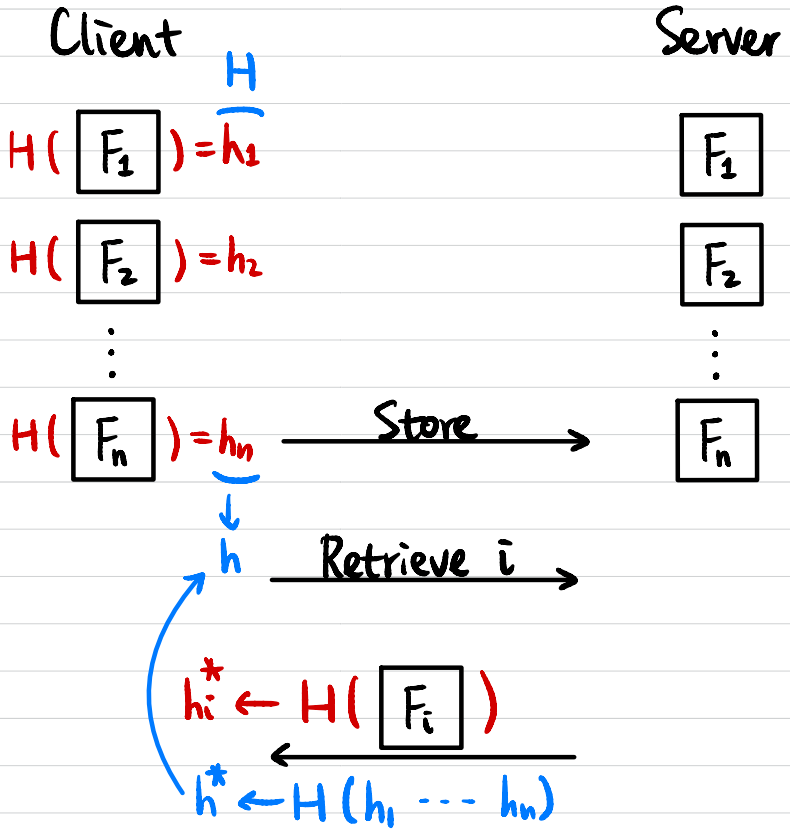
This Lecture:

- Merkle Trees (continued)
- Constructions of Block Cipher
- Substitution-Permutation Network (SPN)
- Feistel Network

Applications of Hash Functions



Is the file changed?



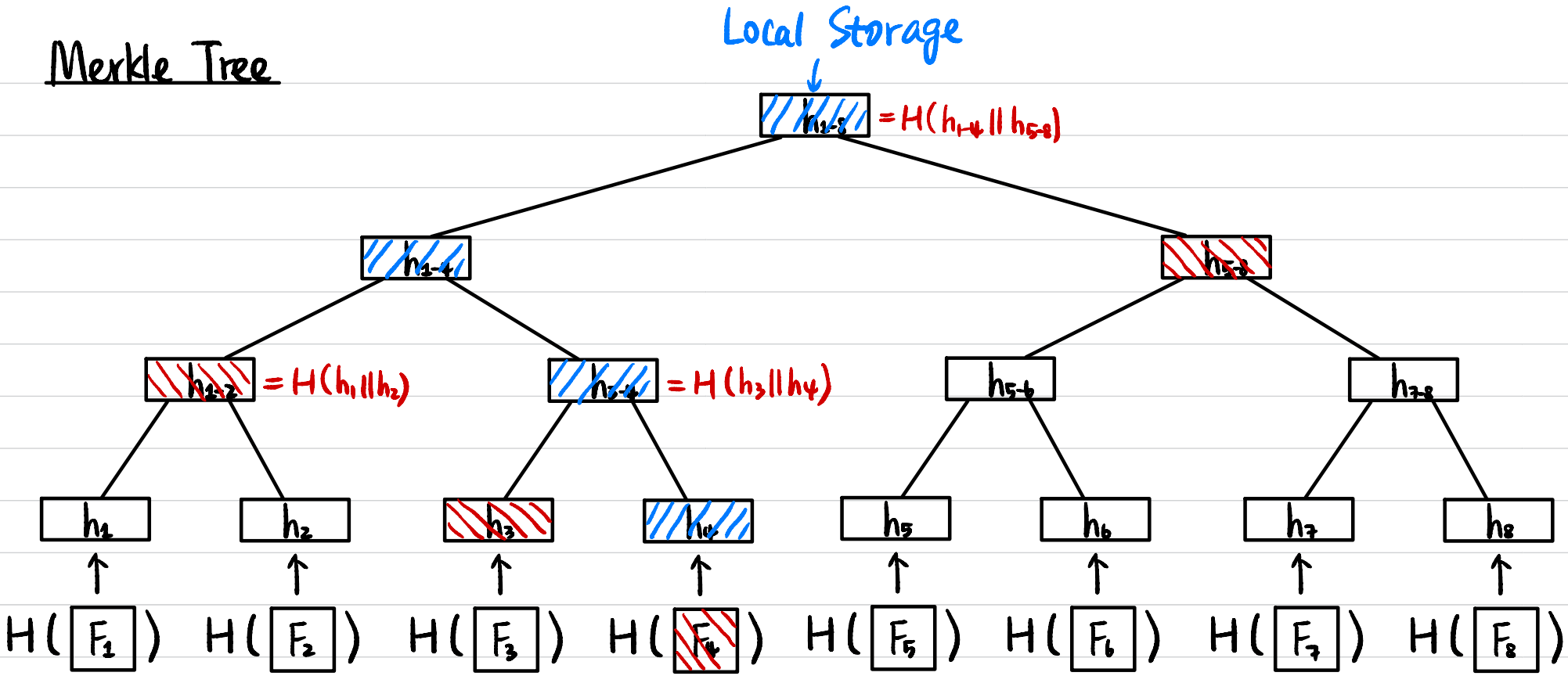
Is the file changed?

Goal:

① Client's storage doesn't grow with n. — O(1)

② Verification doesn't grow with n. — O(log n)

Merkle Tree

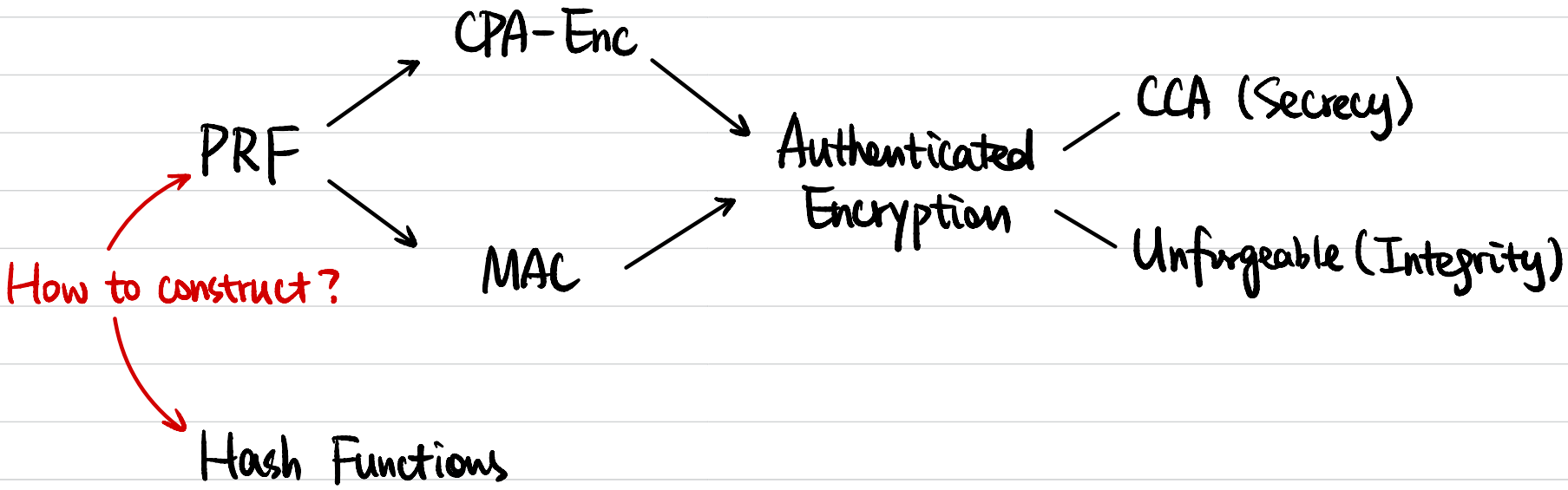


$$H^s : \{0,1\}^* \to \{0,1\}^n$$

$$MT_t^s(F_1 \| \cdots \| F_t) \to \{0,1\}^n$$

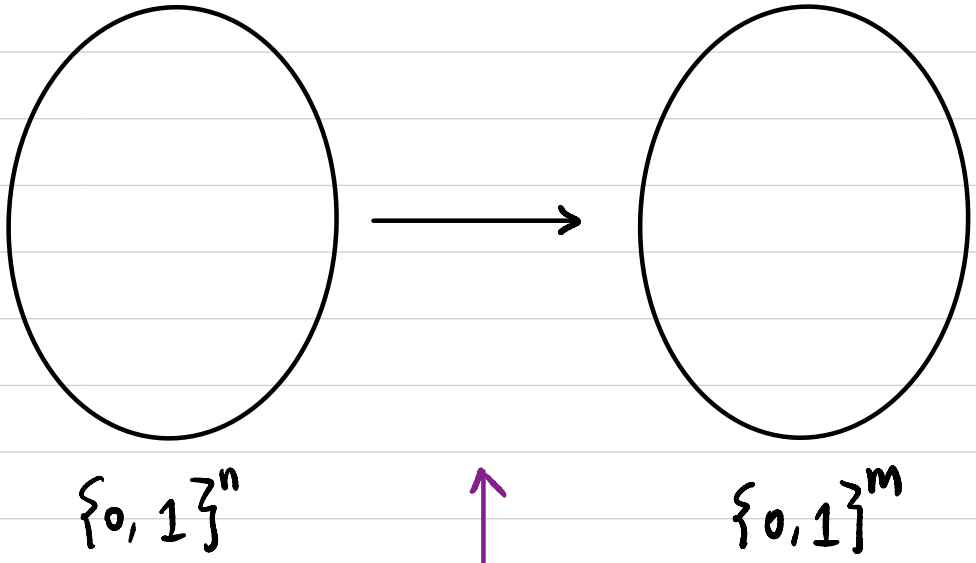
How does verification work?

Thm If (Gen, H) is a CRHF, then (Gen, MT_t) is a CRHF for any fixed $t = 2^k$.


CPA-Enc
PRF
MAC
Authenticated Encryption
CCA (Secrecy)
Unforgeable (Integrity)
How to construct?
Hash Functions

Pseudorandom Function (PRF)

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



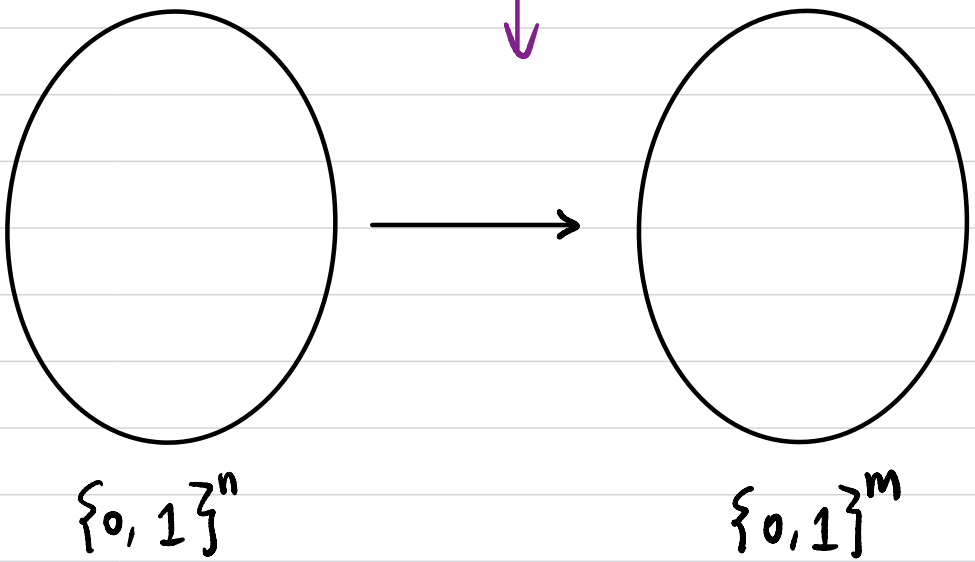
How many possible F_k's ?

2^λ

$\forall$ PPT A
(not knowing k)

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

f:



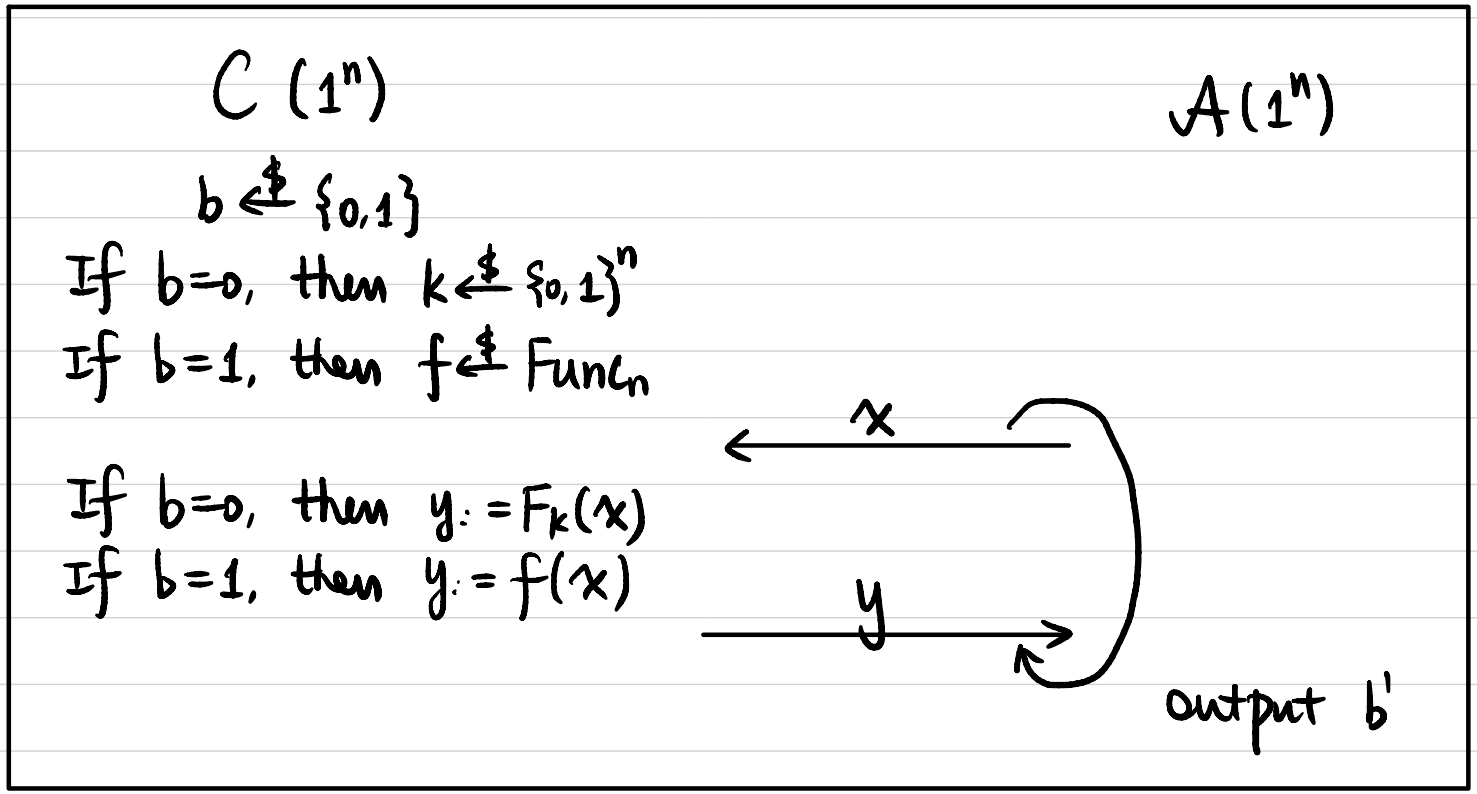
How many possible f's ?

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

Pseudorandom Function (PRF)

Def Let $F: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a deterministic, poly-time, keyed function. F is a pseudorandom function (PRF) if $\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t.

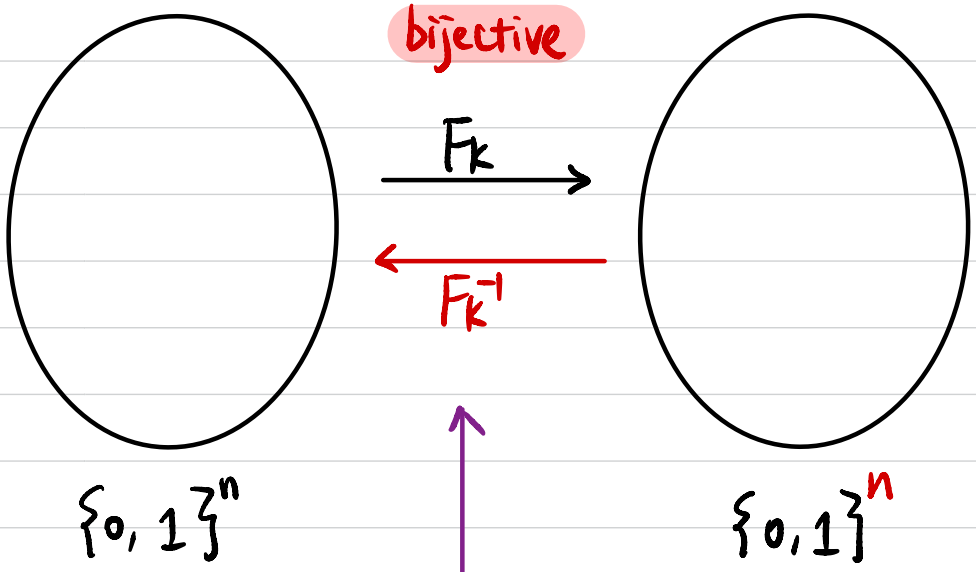
$$\left| \Pr_{k \leftarrow U_n}[A^{F_k(\cdot)}(1^n) = 1] - \Pr_{f \xleftarrow{\$} \text{Func}_n}[A^{f(\cdot)}(1^n) = 1] \right| \leq \varepsilon(n)$$



$$\Pr[b = b'] \leq \frac{1}{2} + \varepsilon(n).$$

Pseudorandom Permutation (PRP)

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

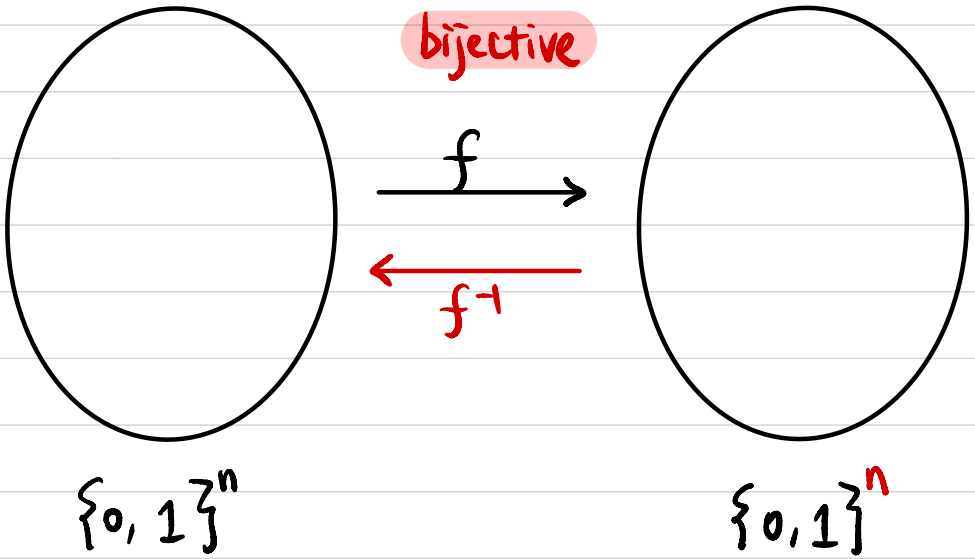


How many possible F_k's ?

$\forall$ PPT A
(not knowing k)

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

$f:$

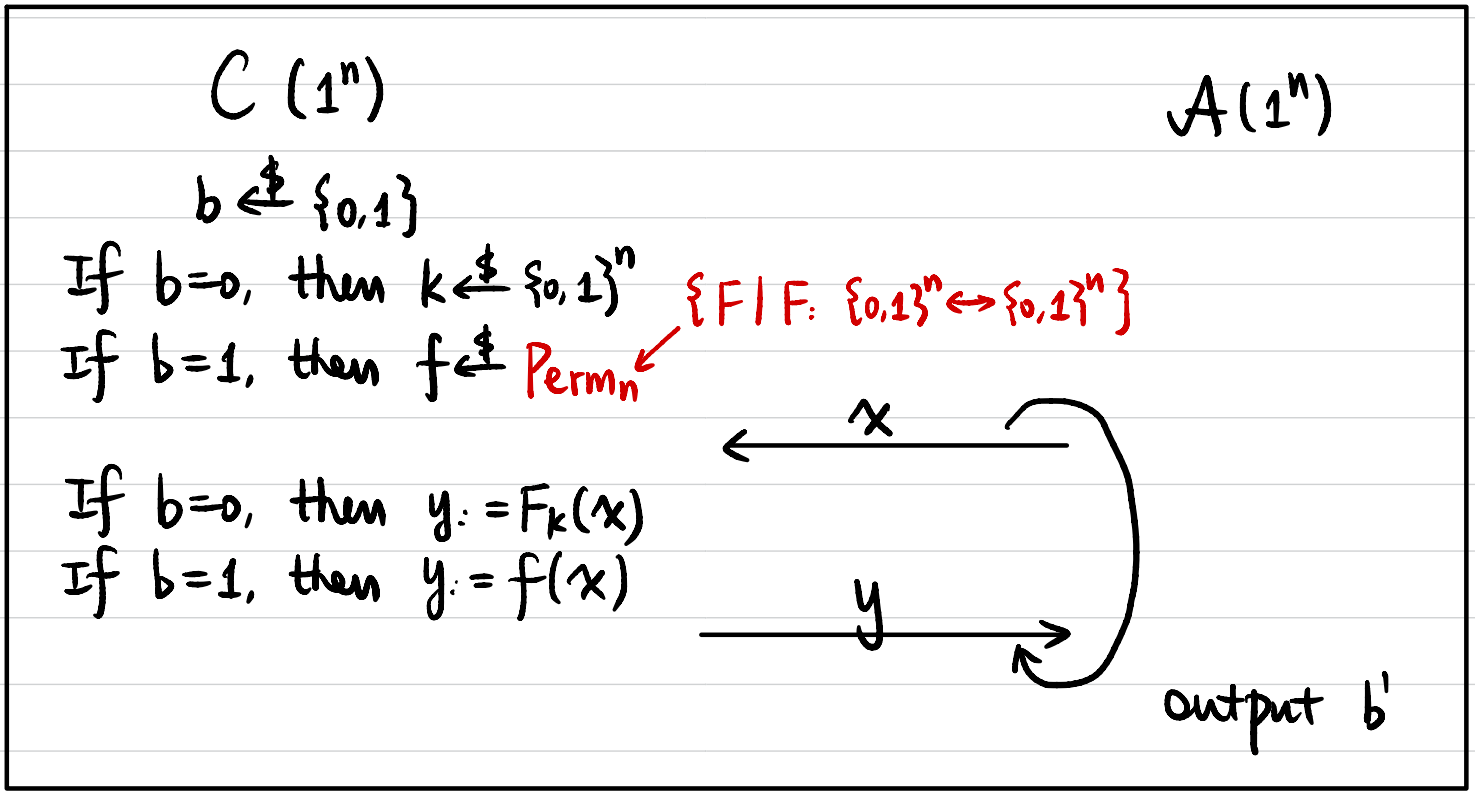


How many possible f's ?

Pseudorandom Permutation (PRP)

Def Let $F: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a deterministic, poly-time, keyed function. F is a **pseudorandom permutation (PRP)** if $F_k(\cdot)$ is bijective for all k, $\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t.

$$\left| \Pr_{k \leftarrow U_n}[A^{F_k(\cdot)}(1^n) = 1] - \Pr_{f \xleftarrow{\$} Perm_n}[A^{f(\cdot)}(1^n) = 1] \right| \leq \varepsilon(n)$$



$$\Pr[b = b'] \leq \frac{1}{2} + \varepsilon(n).$$

Block Cipher

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

n: key length

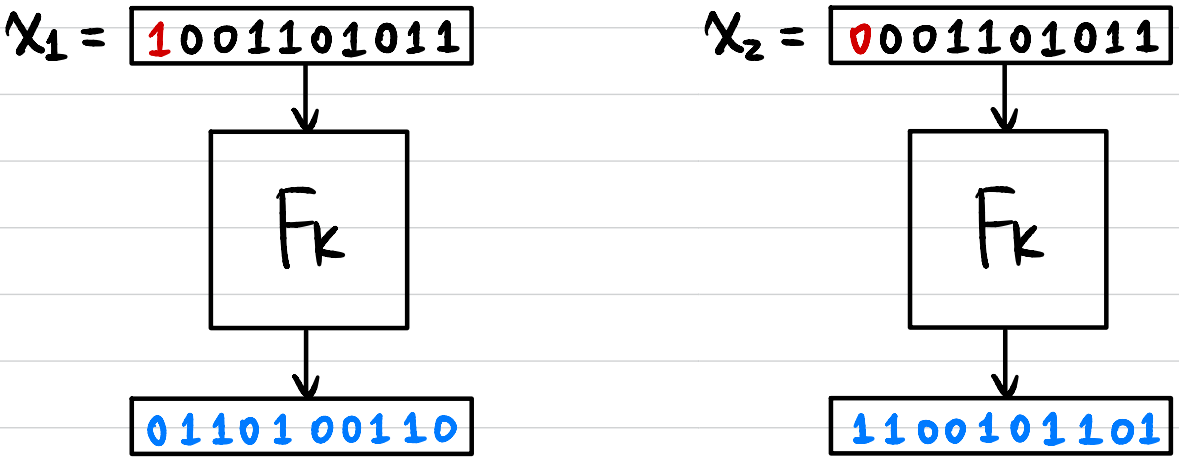
l: block length

$F_k(\cdot)$: permutation / bijective $\{0,1\}^l \to \{0,1\}^l$

$F_k^{-1}(\cdot)$: efficiently computable given k.

Assumed to be a pseudorandom permutation (PRP).

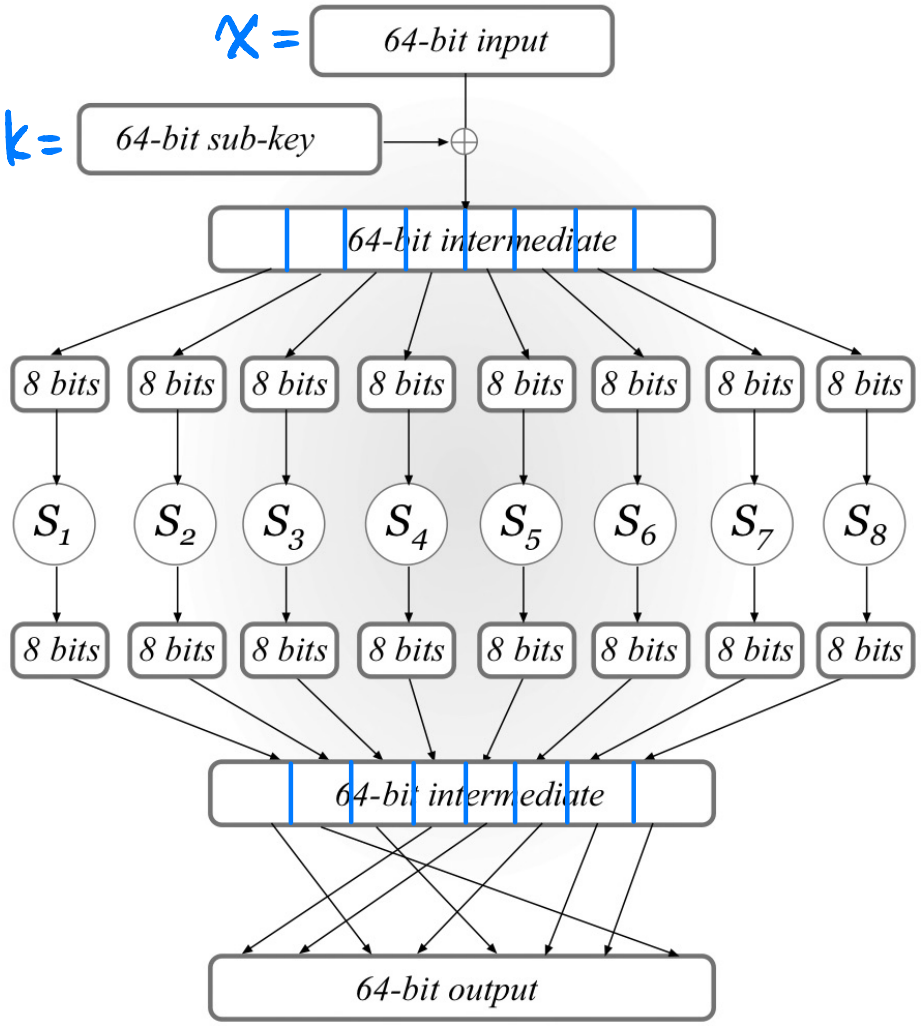
Substitution-Permutation Network (SPN)



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)

$S_i : \{0,1\}^8 \to \{0,1\}^8$ (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)

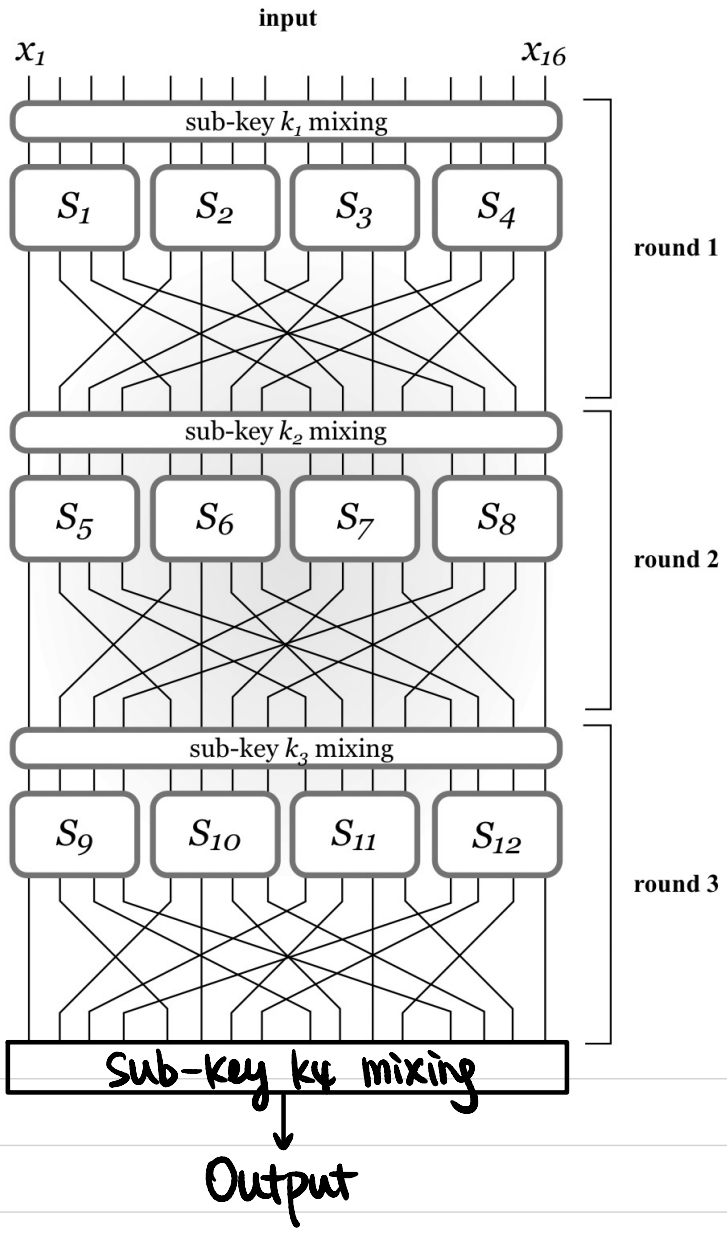
$P : [64] \to [64]$

Public mixing permutation

↓

affect input to multiple S-boxes next round

Substitution-Permutation Network (SPN)



3-round SPN:

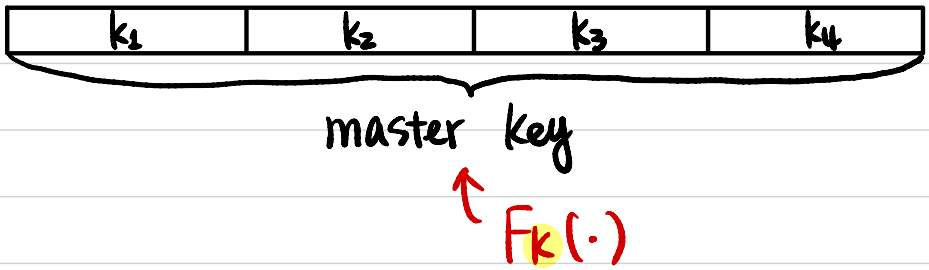
3-round
- key mixing
- substitution
- permutation

1 final-round key mixing

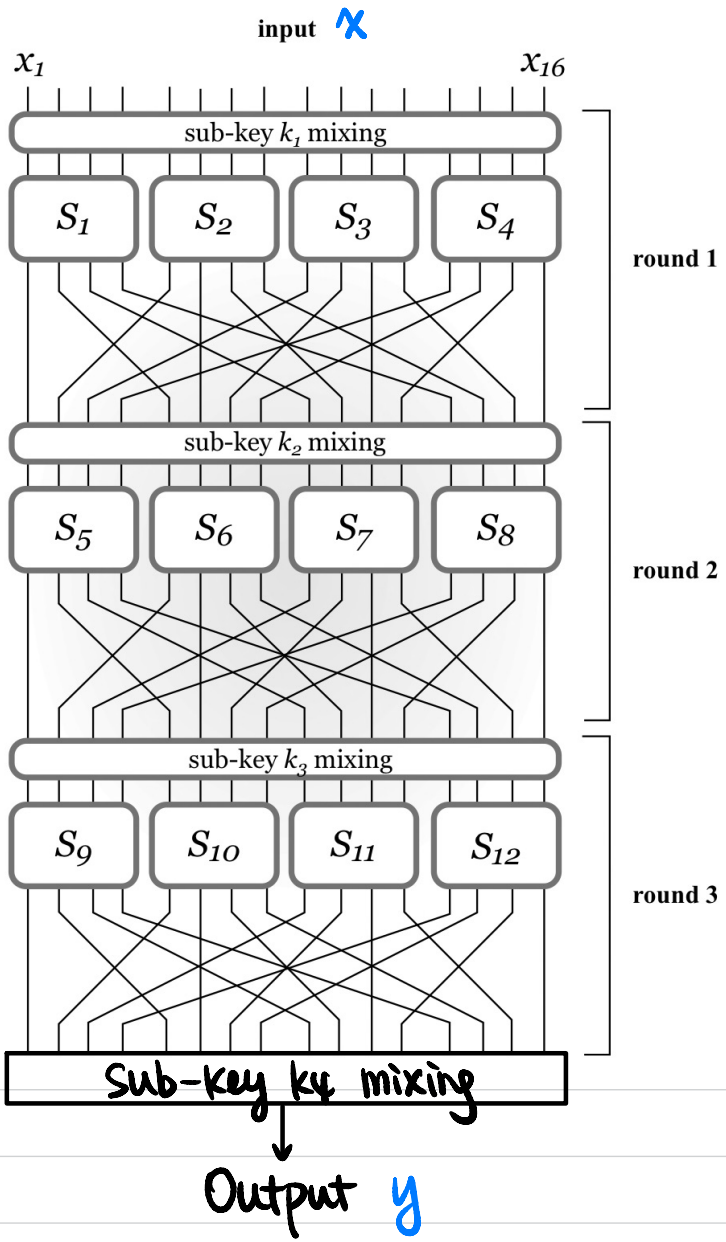
Key Schedule:

How we derive sub-keys from master key.

Example:



Substitution-Permutation Network (SPN)



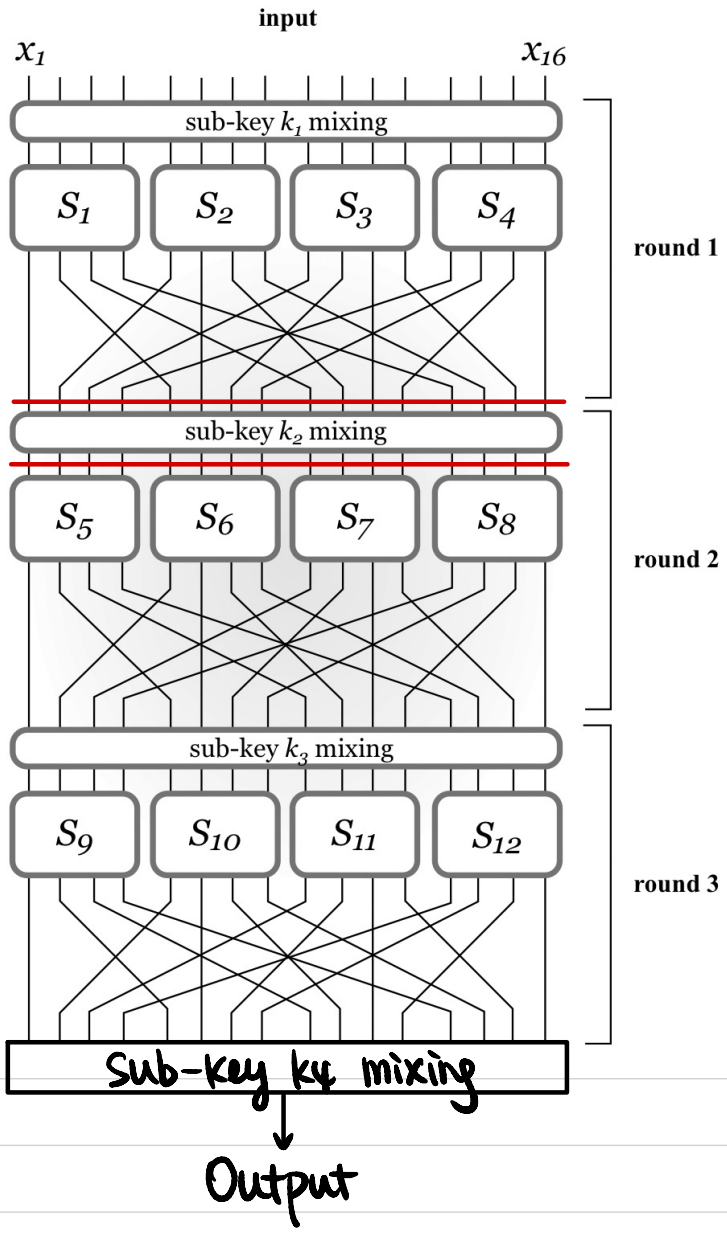
An SPN is invertible given the master key.

⇓

permutation

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

Attacks on Reduced-Round SPN



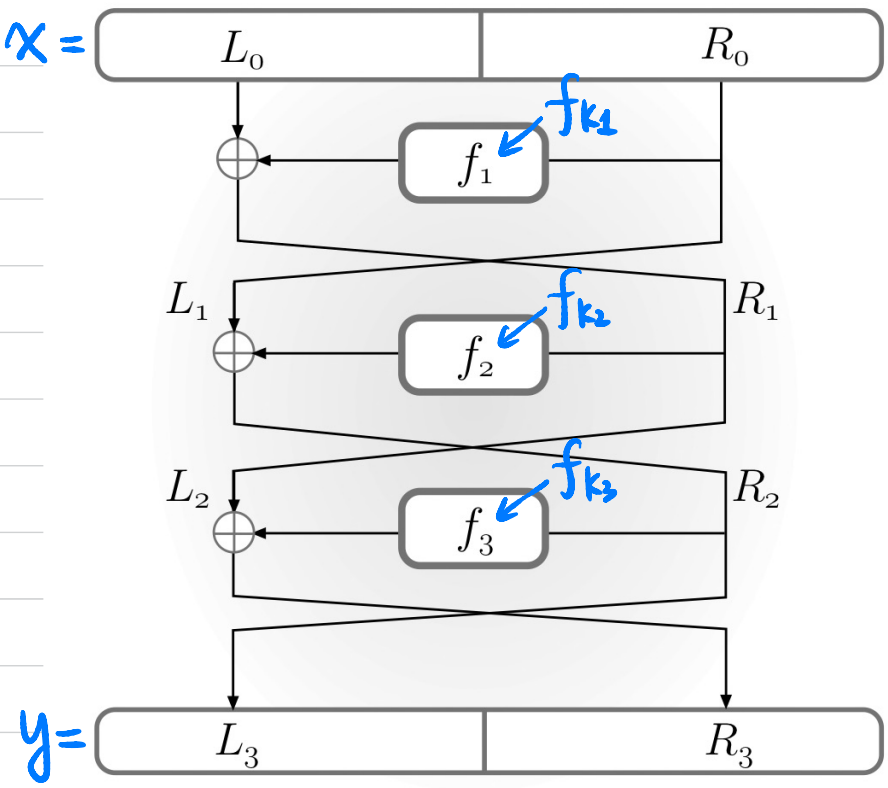
1-round SPN without final key mixing?

1-round SPN with final key mixing?

Why do we need a final key mixing step?

Can we do r-round key mixing, then r-round substitution, then r-round permutation?

Feistel Network



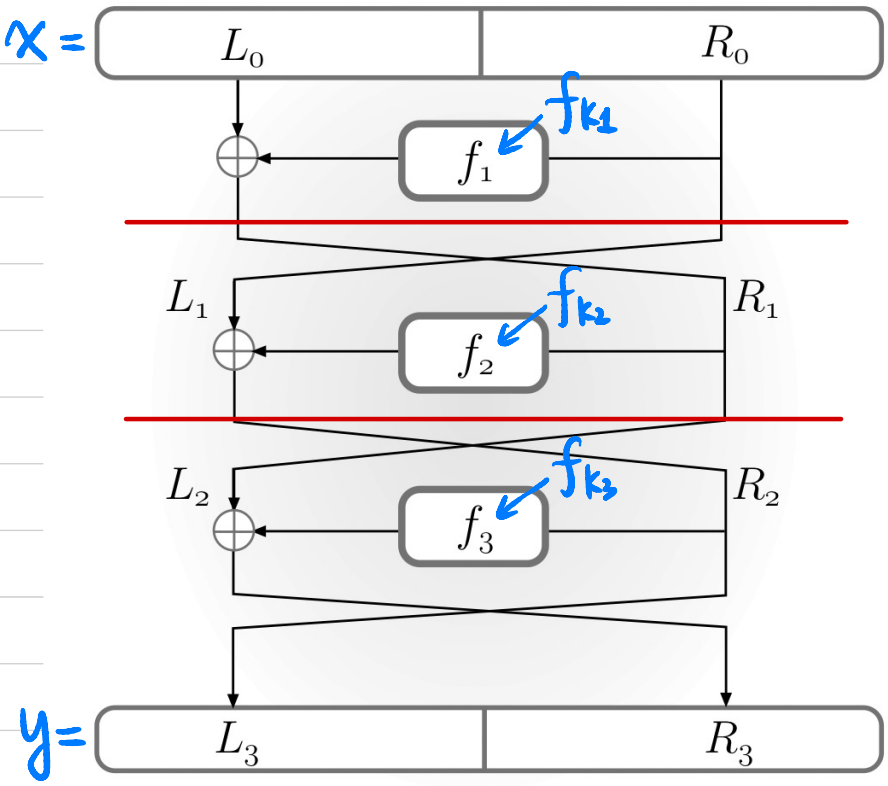
3-round Feistel Network

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

↑ round function

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

Attacks on Reduced-Round Feistel Network



1-round ?

2-round ?