CSCI 1515 Applied Cryptography

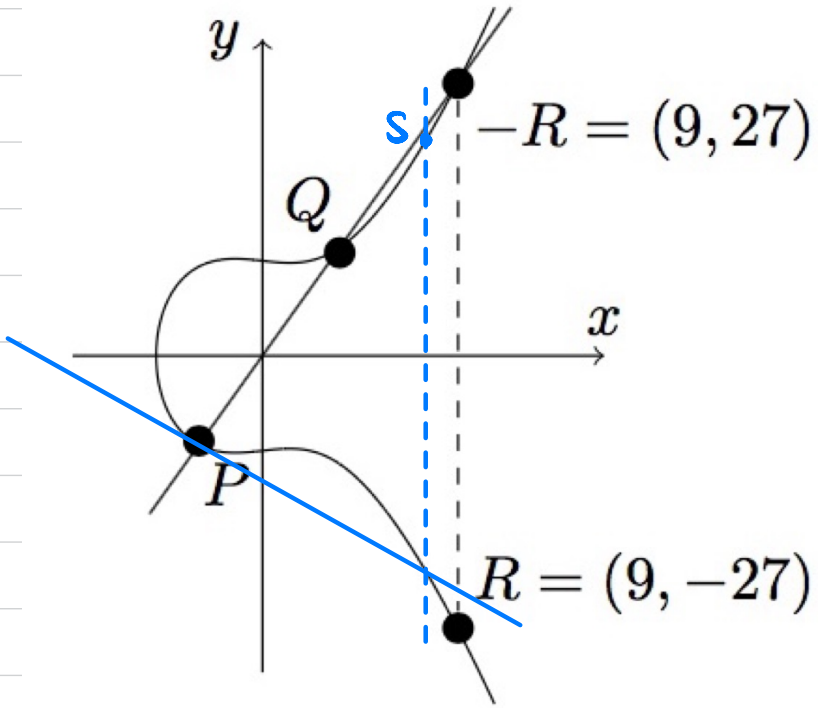
This Lecture:

- Elliptic Curve Cryptography (Continued)
- Practical Constructions of Block Cipher

Elliptic Curves

Example: $y^2 = x^3 - x + 9$

How to find rational points $(x, y) \in \mathbb{Q}^2$ on the curve?



① Chord method $\quad R := P \boxplus Q$

$P = (-1, -3)$
$Q = (1, 3)$
$\Rightarrow \quad y = 3x$

$\Downarrow$

$(3x)^2 = x^3 - x + 9$

$x^3 - 9x^2 - x + 9 = 0$

Why is the third root rational?

$(x - x_1)(x - x_2)(x - x_3) = 0$

$x_1 = -1, \quad x_2 = 1$

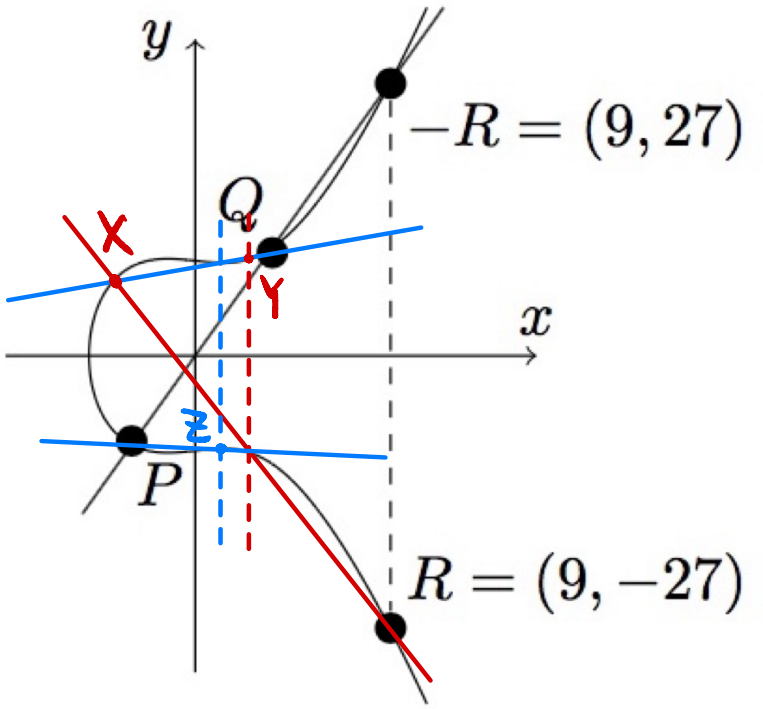
$(-x_1)(-x_2)(-x_3) = 9$

$x_3 = \dfrac{-9}{x_1 \cdot x_2} = 9$

② tangent method $\quad S := P \boxplus P$

Elliptic Curves

Example: $y^2 = x^3 - x + 9$



$R := P \boxplus Q$

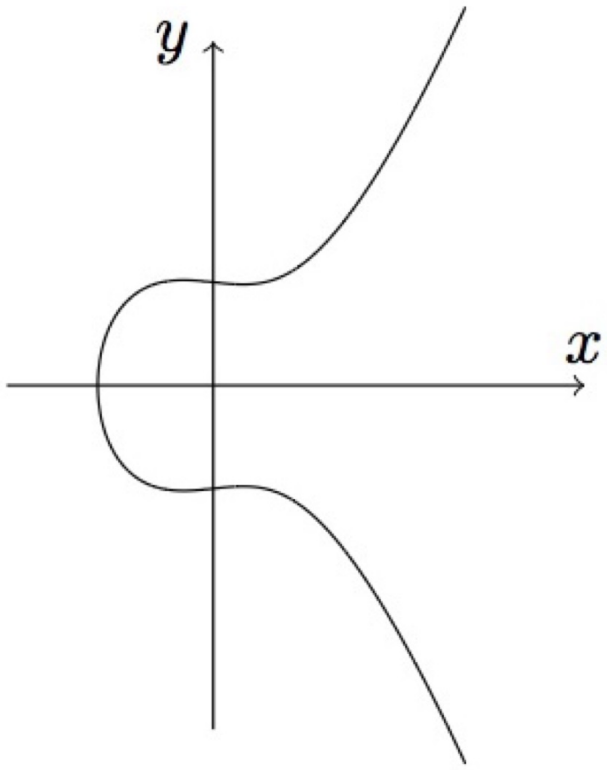
$P \boxplus Q = Q \boxplus P$

$(P \boxplus Q) \boxplus X = P \boxplus (Q \boxplus X)$

$R = P \boxplus Q$ $\quad Z = Q \boxplus X$

$Y = R \boxplus X$ $\quad Y = P \boxplus Z$

Elliptic Curves over Finite Fields



$y^2 = x^3 + ax + b$

$(4a^3 + 27b^2 \neq 0)$

Finite field $\mathbb{F}_p$, $p > 3$ prime ($\{0, 1, \cdots, p-1\}$, $+$, $\cdot$, inverse)

Elliptic cure E defined over $\mathbb{F}_p$: $E/\mathbb{F}_p$.

$a, b \in \mathbb{F}_p$

(x, y) is a point on the cure if

$x, y \in \mathbb{F}_p$

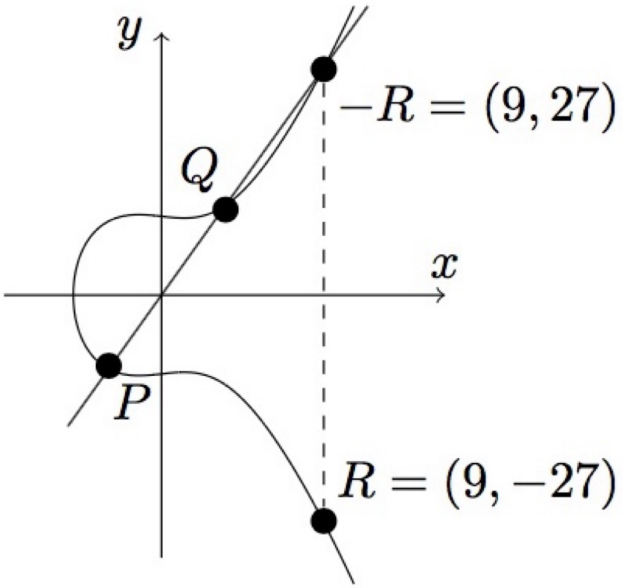
$y^2 = x^3 + ax + b$ over $\mathbb{F}_p$

Point at infinity: O

Example: $y^2 = x^3 + 1$ over $\mathbb{F}_{11}$.

$E/\mathbb{F}_{11} = \{O, (-1, 0), (0, \pm 1), (2, \pm 3), (5, \pm 4), (7, \pm 5), (9, \pm 2)\}$

Elliptic Curves over Finite Fields



Group properties:

Why is it well-defined?

$\boxplus$

① Closure: $\forall g, h \in G,\ g \circ h \in G$ $\quad R \boxplus R := O$

② Existence of an identity: $\quad O \boxplus R := R$

$\exists e \in G$ s.t. $\forall g \in G,\ e \circ g = g \circ e = g.$

③ Existence of inverse: $\quad \text{inv}(R) = -R$

$\forall g \in G,\ \exists h \in G$ s.t. $g \circ h = h \circ g = e$

④ Associativity:

$\forall g_1, g_2, g_3 \in G,\ (g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$

⑤ Commutativity (abelian):

$\forall g, h \in G,\ g \circ h = h \circ g$

SEA algorithm: count number of points on $E/\mathbb{F}_p$ in time $\text{polylog}(p)$.

How to compute g^a for $a \in \mathbb{Z}_q$?

$\underbrace{g \boxplus g \boxplus \cdots \boxplus g}_{a}$

$g \boxplus g = g^2$

$g^2 \boxplus g^2 = g^4$

$g^4 \boxplus g^4 = g^8$

Elliptic Curve Cryptography

- Curve secp256r1 (P256)
  - prime $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$
  - $y^2 = x^3 - 3x + b$ b: 255-bit
  - Number of points on the curve is prime (close to p)
  - Generator point G

- Curve secp256k1
- Curve 25519

Summary

	Symmetric-Key	Public-Key
Message Secrecy	Primitive: SKE Construction: block cipher	Primitive: PKE Constructions: RSA / ElGamal
Message Integrity	Primitive: MAC Constructions: CBC-MAC / HMAC	Primitive: Signature Constructions: RSA / DSA
Secrecy & Integrity	Primitive: AE Construction: Encrypt-then-MAC	
Key Exchange		Construction: Diffie-Hellman
Important Tool	Primitive: Hash function Construction: SHA	

Block Cipher

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

λ: key length

n: block length

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

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

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

Construction: Advanced Encryption Standard (AES)

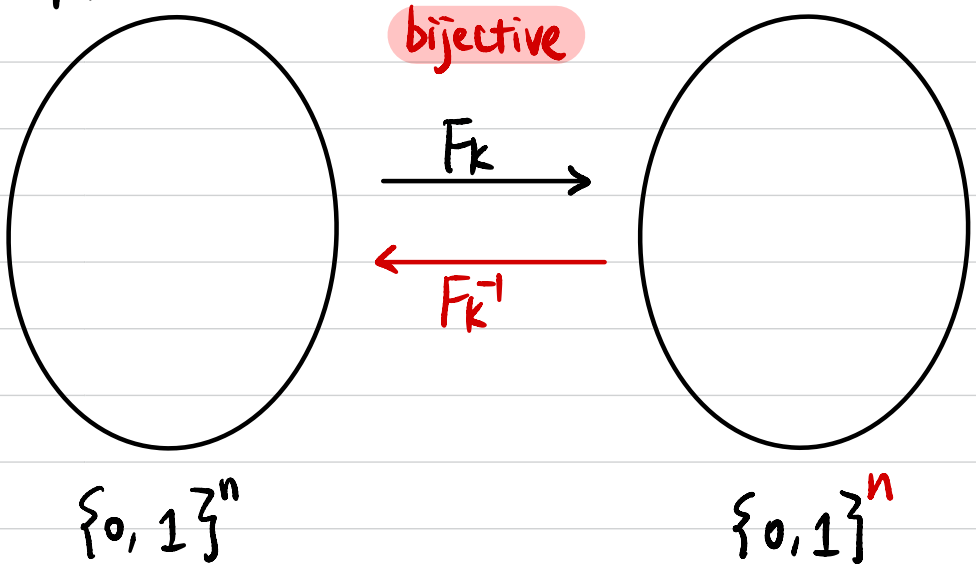
- $\lambda = 128/192/256$, $n = 128$

Before AES: Data Encryption Standard (DES)

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

Pseudorandom Permutation (PRP)

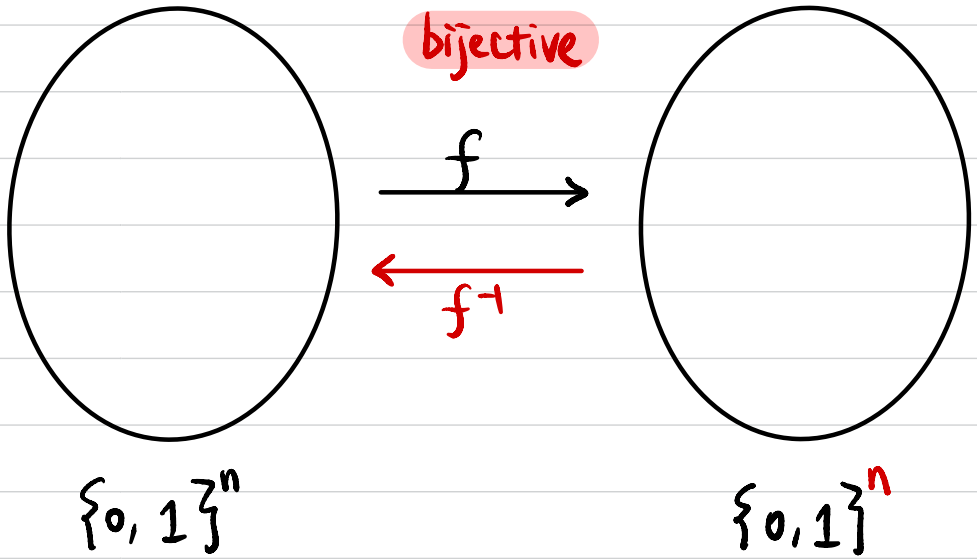
$k \xleftarrow{\$} \{0,1\}^\lambda$ $\quad F_k:$



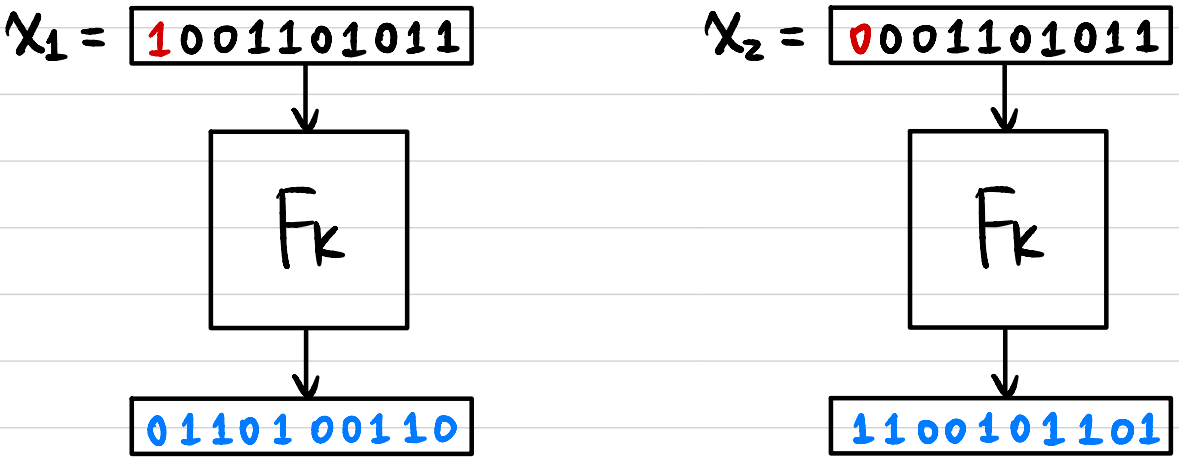
$\{\{\{C$ (not knowing k)

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

$f:$



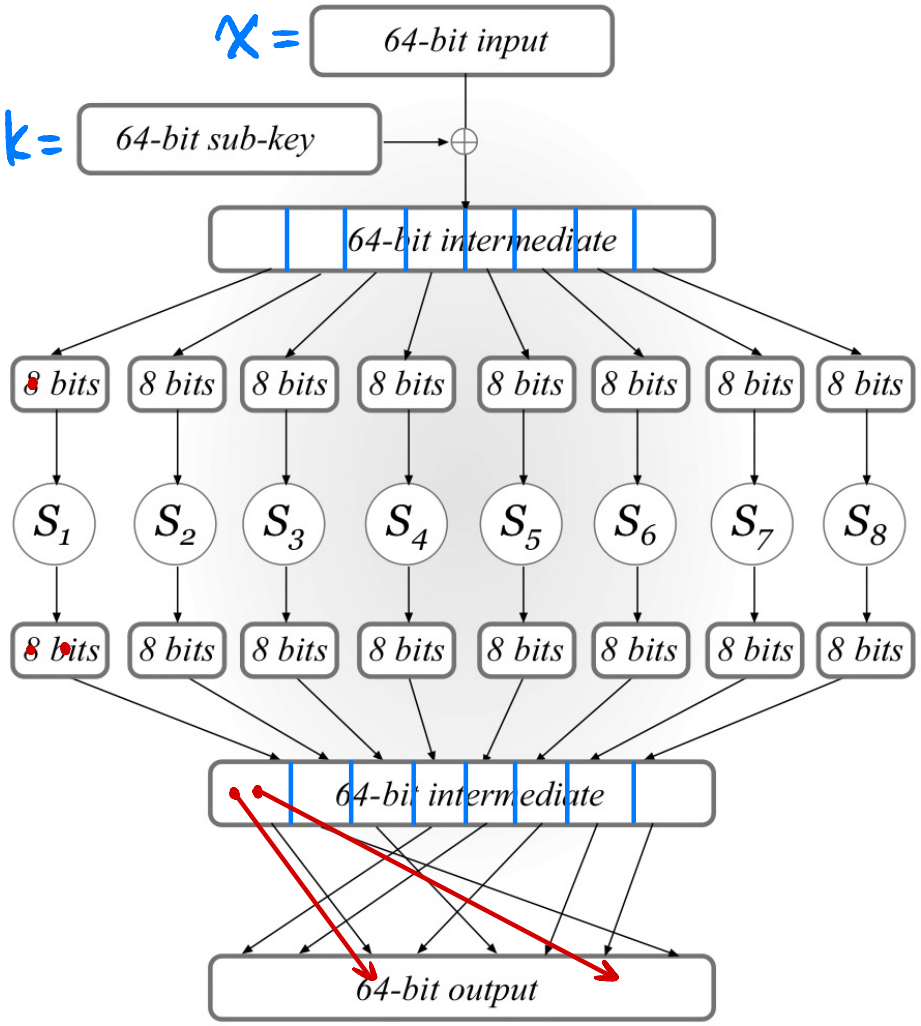
Substitution-Permutation Network (SPN)



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)

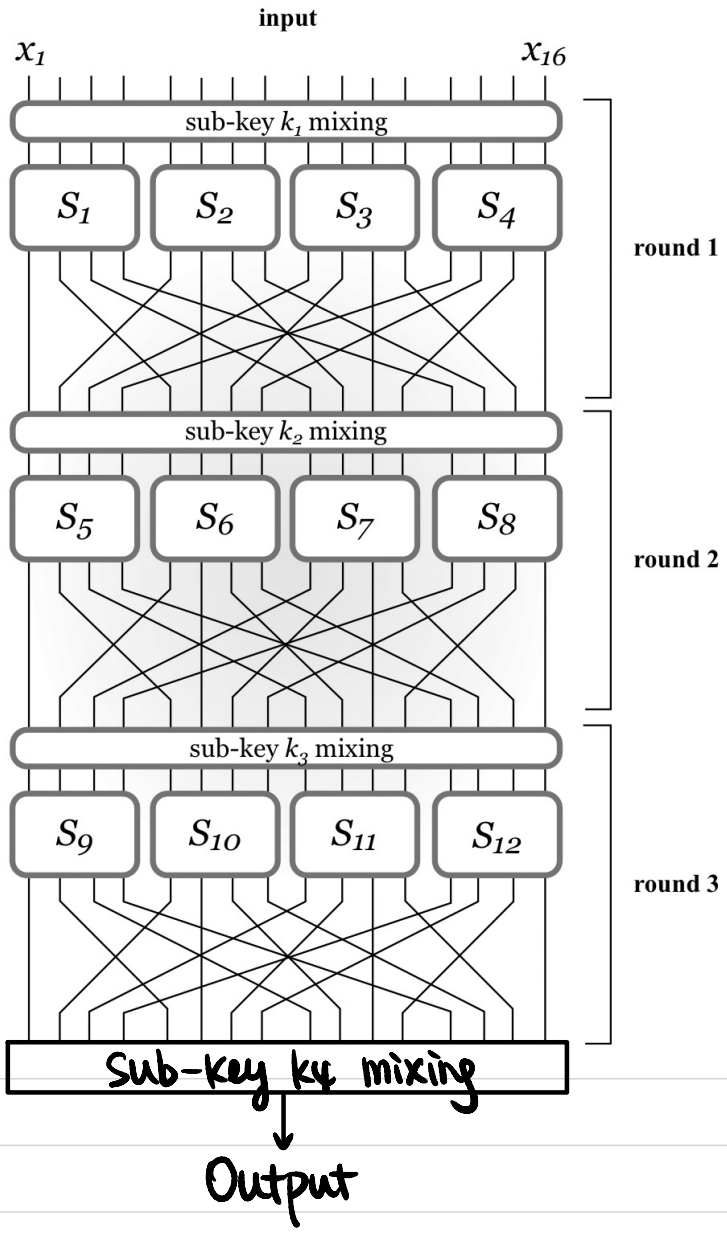
$P : [64] \to [64]$

Public mixing permutation

↓

affect input to multiple S-boxes next round

Substitution-Permutation Network (SPN)



3-round SPN:

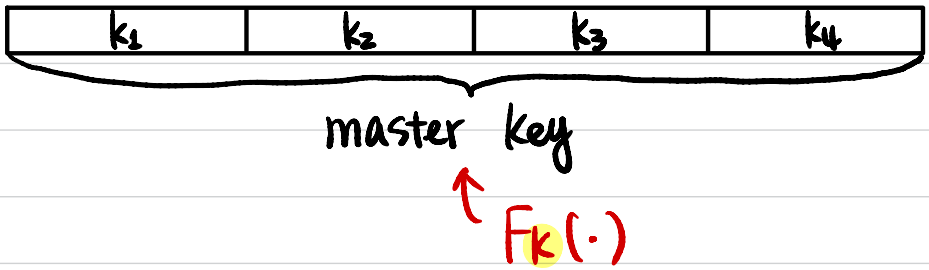
3-round
- key mixing
- substitution
- permutation

1 final-round key mixing

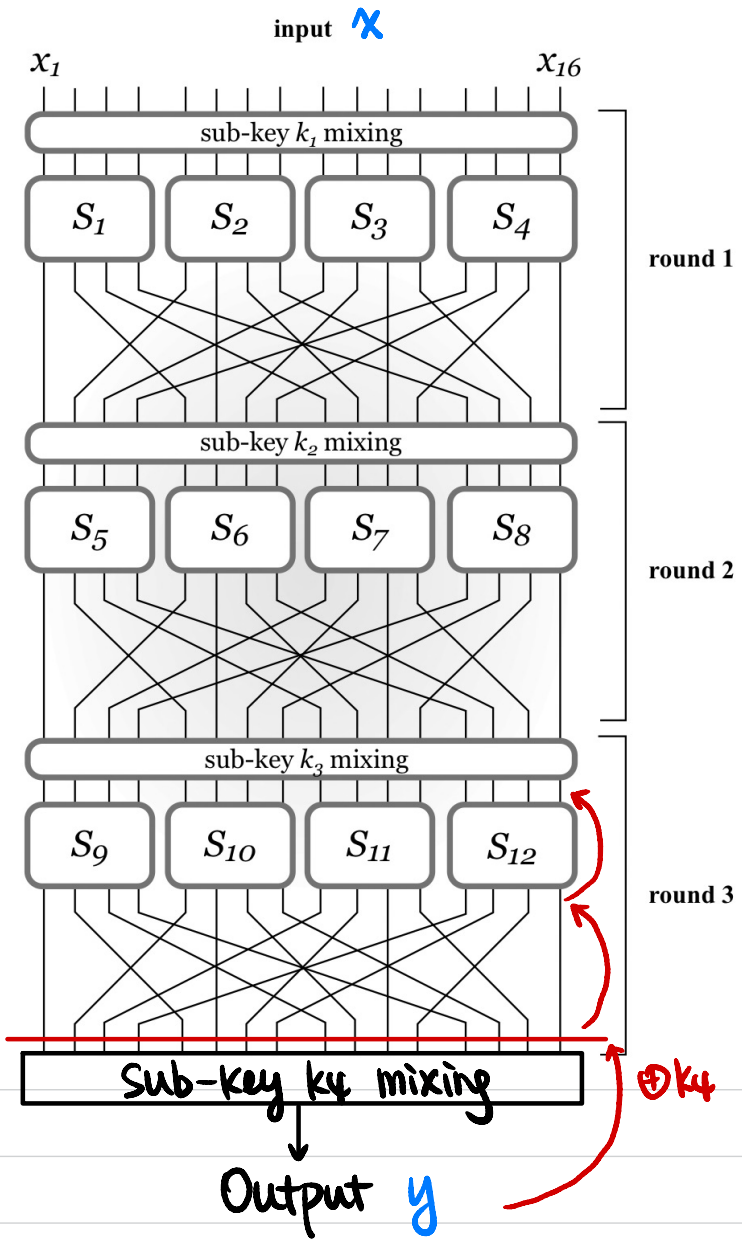
key Schedule:

How we derive sub-keys from master key.

Example:



Substitution-Permutation Network (SPN)



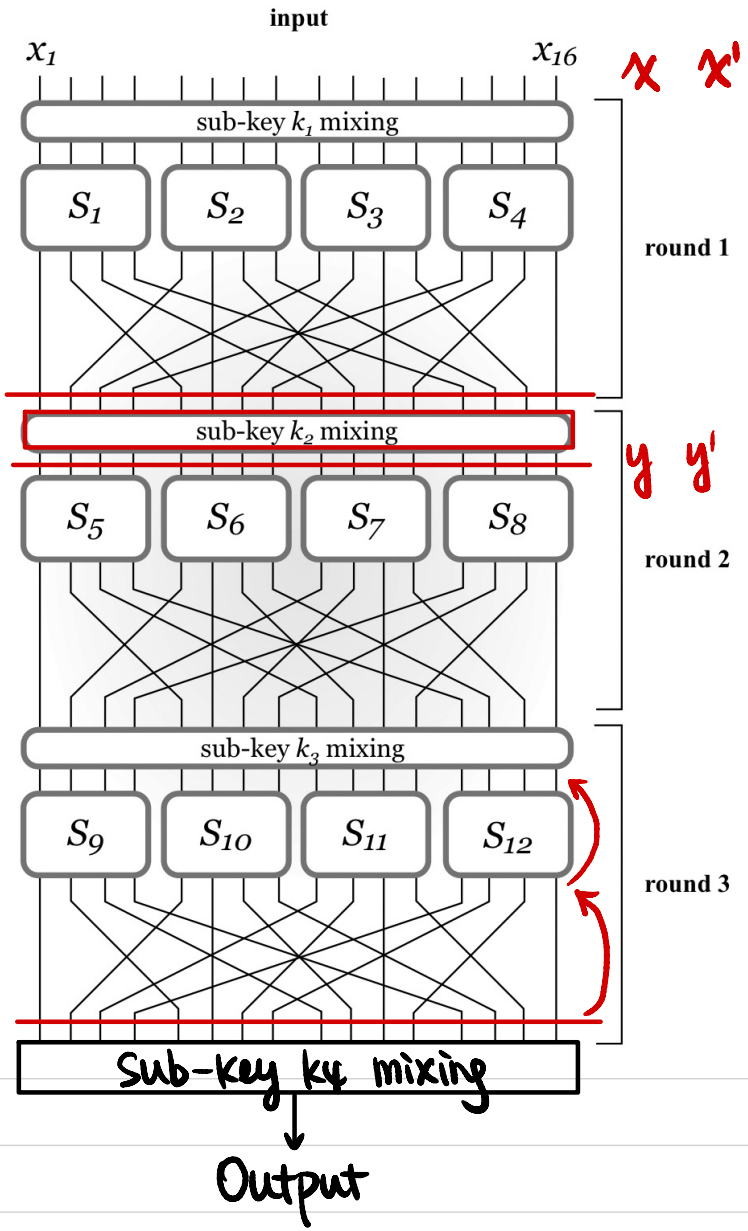
An SPN is invertible given the master key.

⇓

permutation

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

Attacks on Reduced-Round SPN



x x'

y y'

1-round SPN without final key mixing?

Given (x, y) ⇒ recover key?

1-round SPN with final key mixing?

Given (x, y), (x', y') ⇒ recover key?

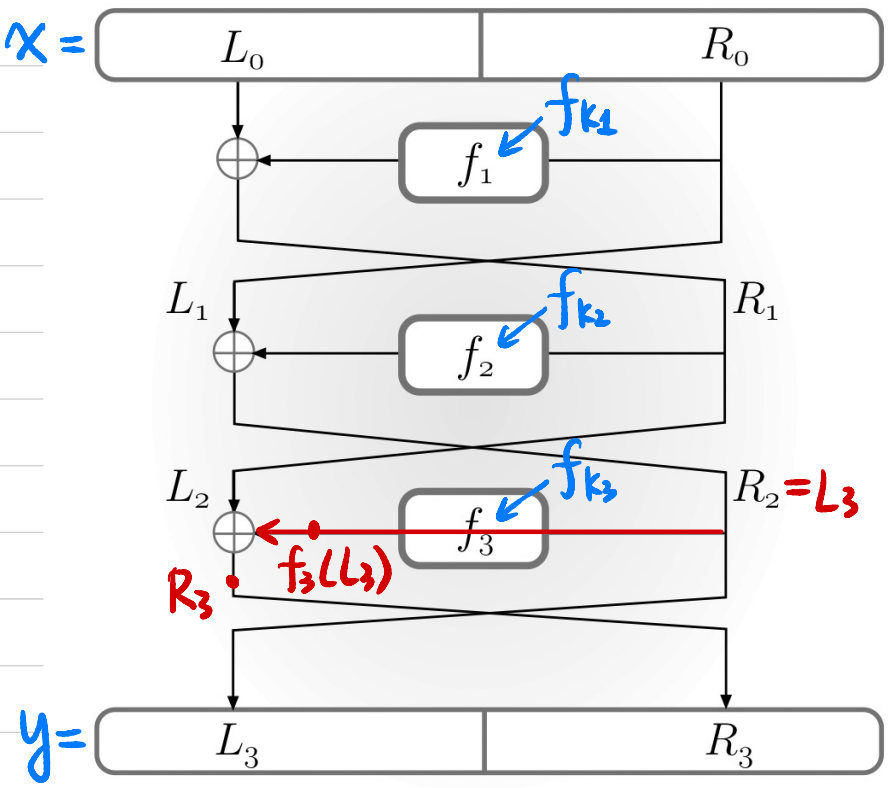
All possible k_2 ⇒ derive k_1

$O(2^{16})$

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

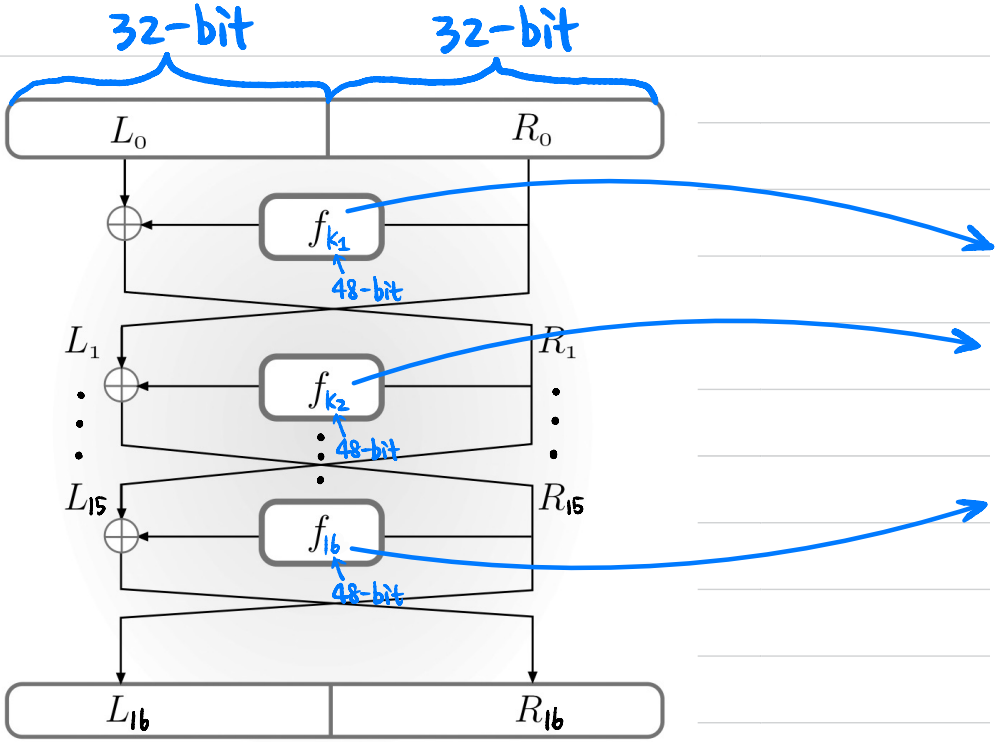
Data Encryption Standard (DES)

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

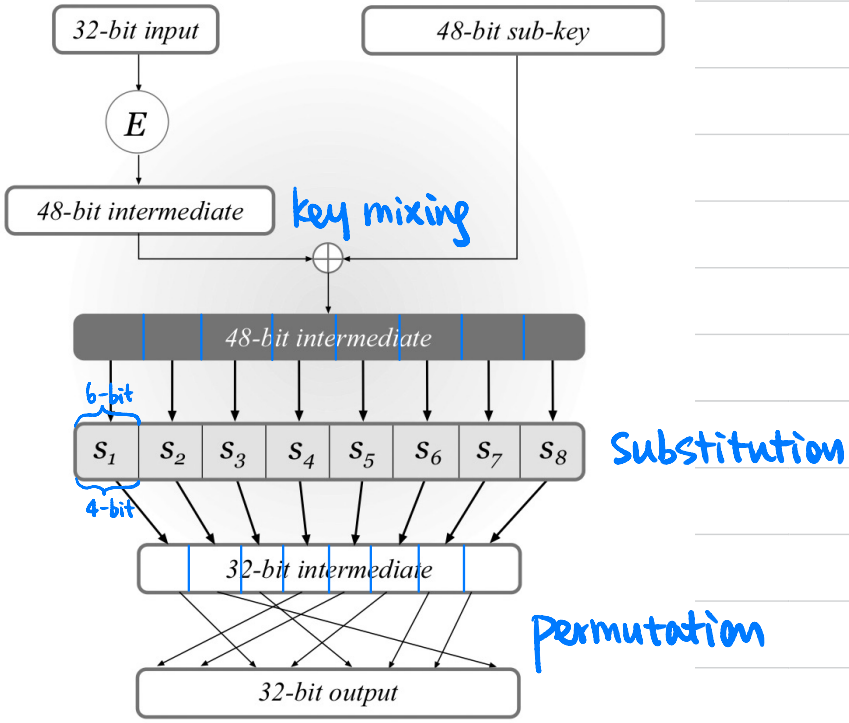
block length $n=64$

master key length $\lambda=56$

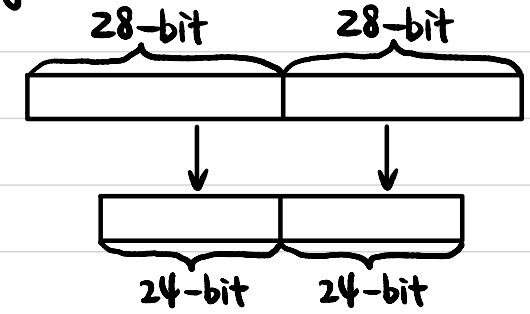
16-round Feistel Network



DES mangler function

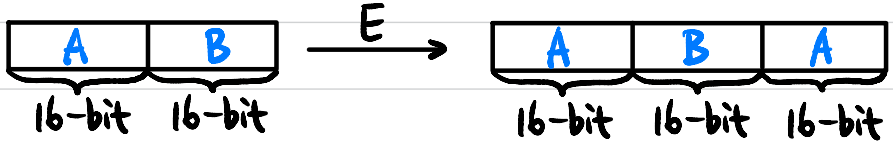


Key Schedule:



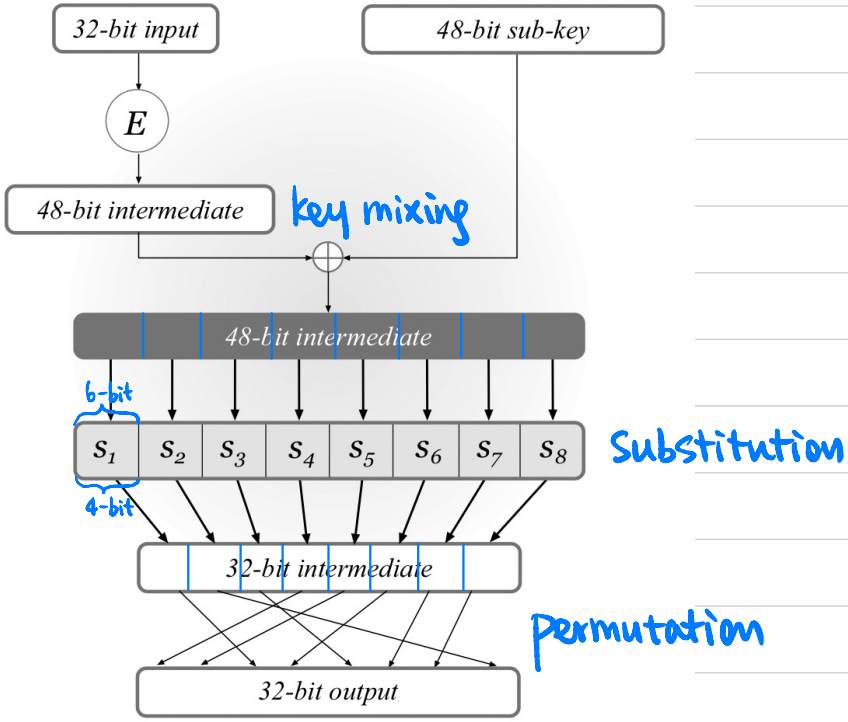
(master key)

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] → [32]

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

Thank You ^^