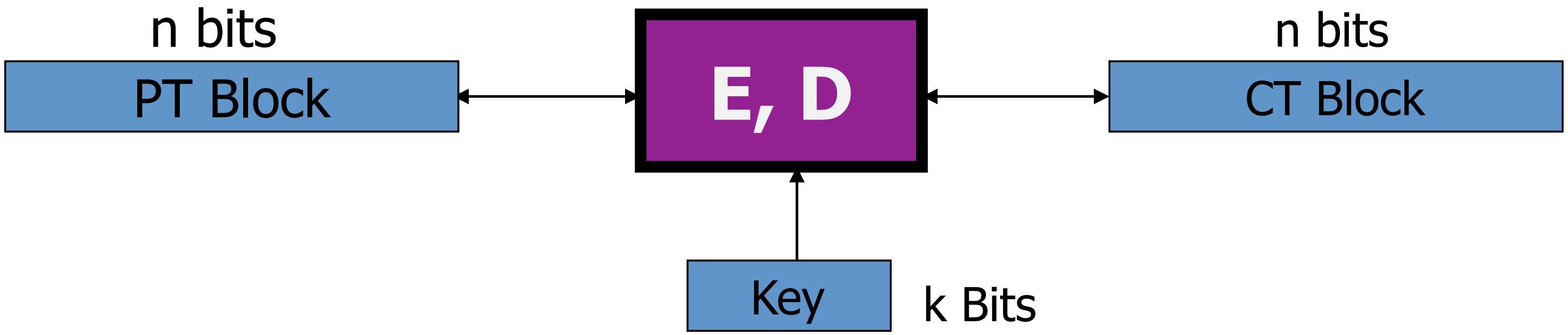
Block ciphers

The data encryption standard (DES)

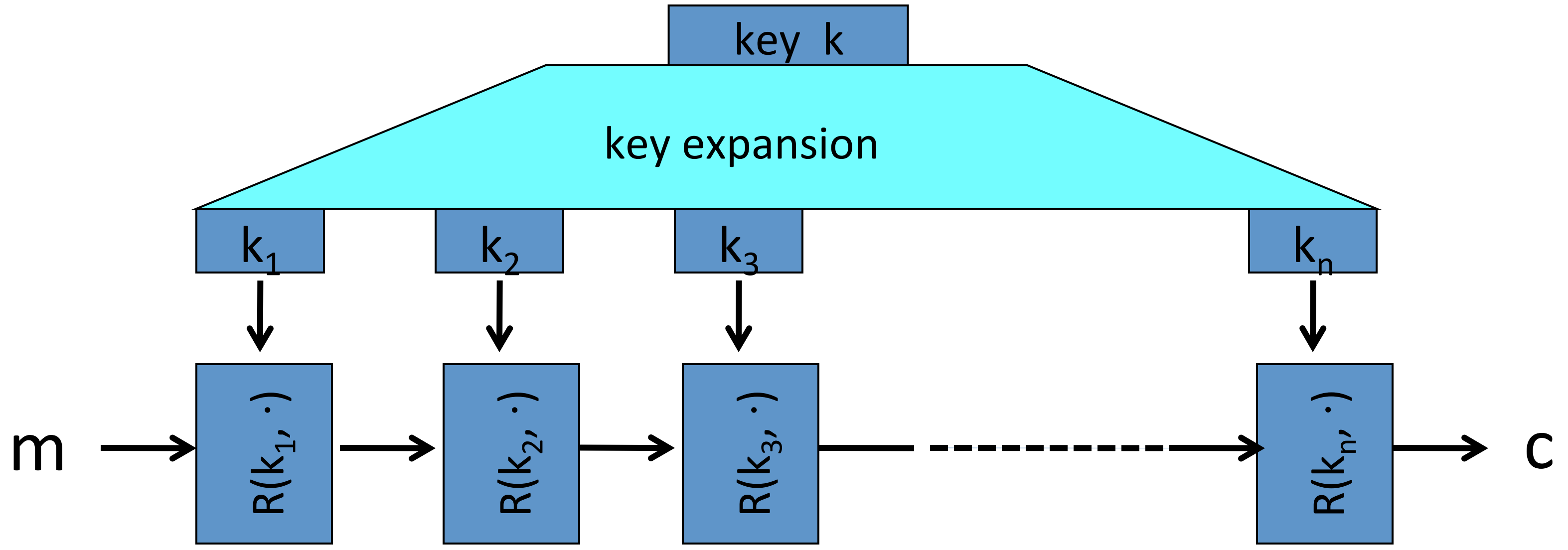
Block ciphers: crypto work horse



Canonical examples:

1. 3DES: n= 64 bits, k = 168 bits
2. AES: n=128 bits, k = 128, 192, 256 bits

Block Ciphers Built by Iteration



$R(k,m)$ is called a round function

for 3DES (n=48), for AES-128 (n=10)

The Data Encryption Standard (DES)

- Early 1970s: Horst Feistel designs Lucifer at IBM

  key-len = 128 bits ; block-len = 128 bits

- 1973: NBS asks for block cipher proposals.

  IBM submits variant of Lucifer.

- 1976: NBS adopts DES as a federal standard

  key-len = 56 bits ; block-len = 64 bits

- 1997: DES broken by exhaustive search

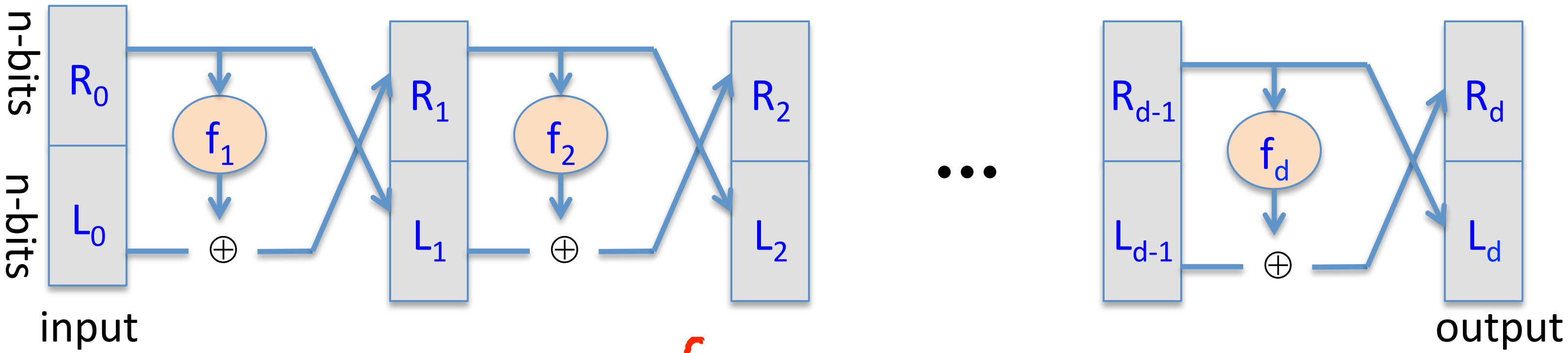
- 2000: NIST adopts Rijndael as AES to replace DES

Widely deployed in banking (ACH) and commerce

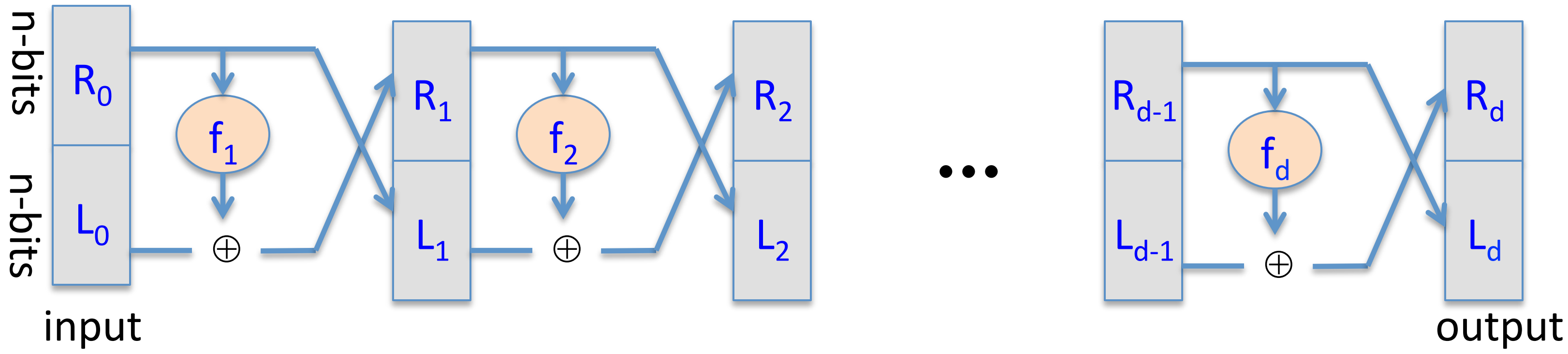
DES: core idea – Feistel Network

Given functions $f_1, \ldots, f_d: \{0,1\}^n \longrightarrow \{0,1\}^n$

Goal: build invertible function $F: \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$



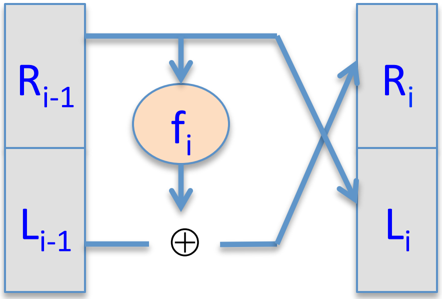
In symbols: $\begin{cases} R_i = f_i(R_{i-1}) \oplus L_{i-1} \\ L_i = R_{i-1} \end{cases}$



Claim: for all $f_1, \ldots, f_d$: $\{0,1\}^n \longrightarrow \{0,1\}^n$

Feistel network $F: \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ is invertible

Proof: construct inverse

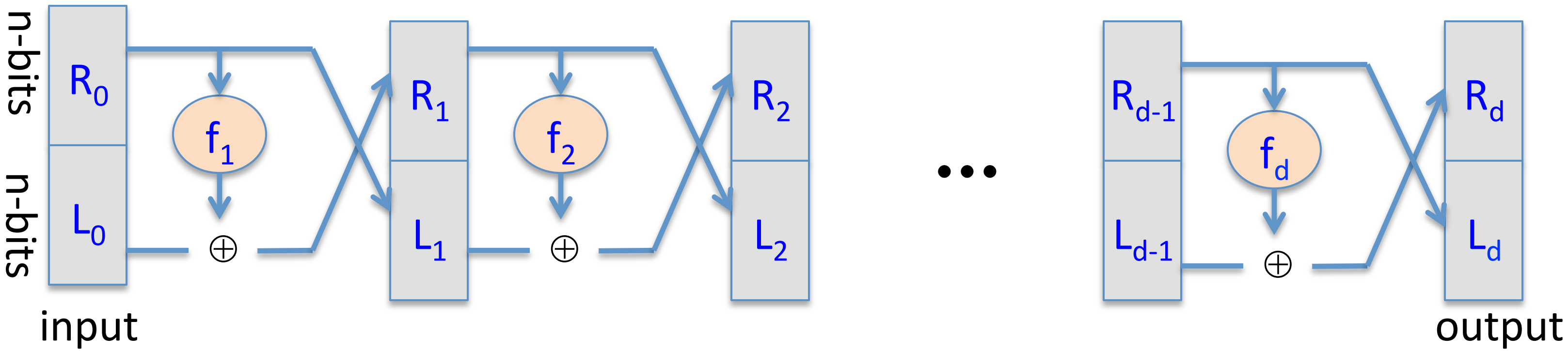


inverse

$R_{i-1} = L_i$

$L_{i-1} =$

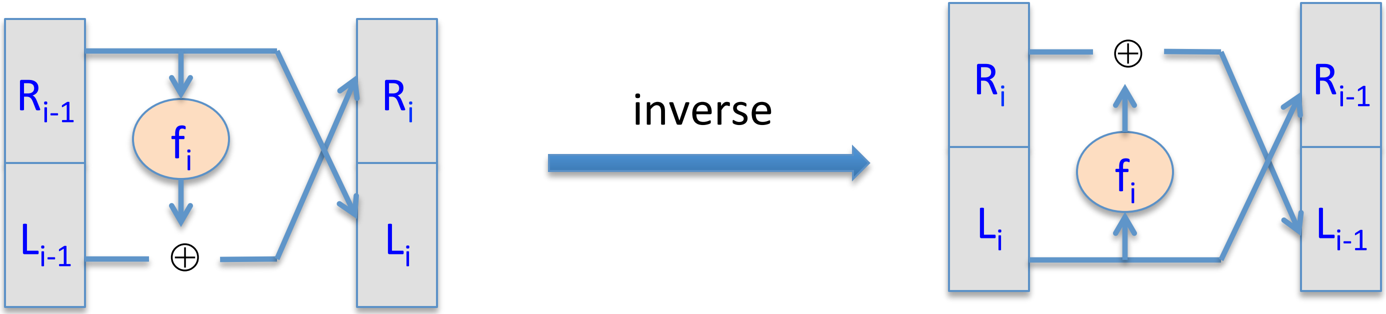
Dan Boneh



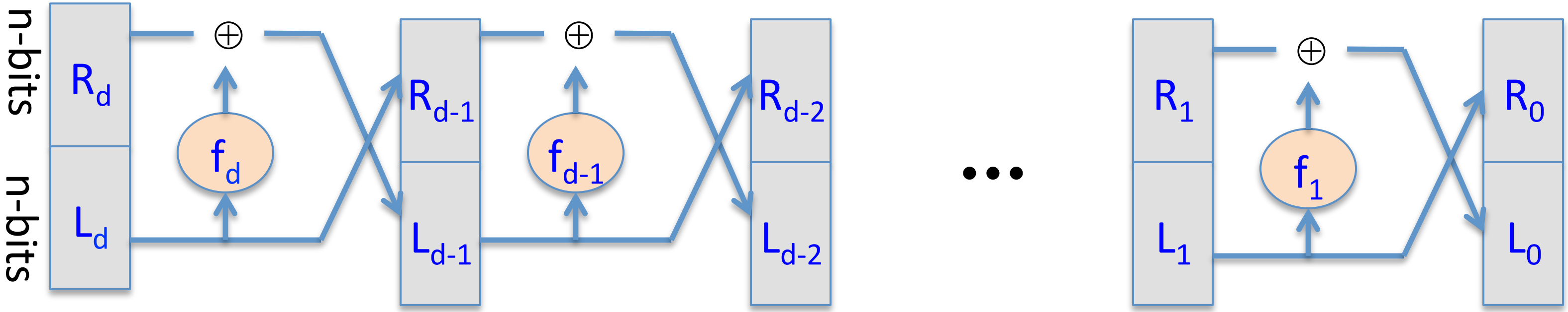
Claim: for all $f_1, \ldots, f_d: \{0,1\}^n \longrightarrow \{0,1\}^n$

Feistel network $F: \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ is invertible

Proof: construct inverse



Decryption circuit



- Inversion is basically the same circuit,
  with $f_1, \ldots, f_d$ applied in reverse order

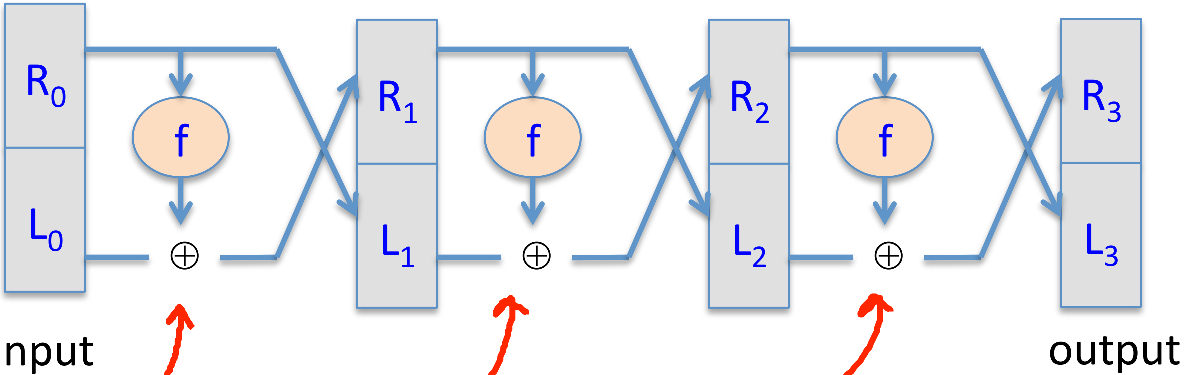
- General method for building invertible functions (block ciphers) from arbitrary functions.

- Used in many block ciphers … but not AES

“Thm:” (Luby-Rackoff ‘85):

$f: K \times \{0,1\}^n \longrightarrow \{0,1\}^n$ a secure PRF

$\Rightarrow$ 3-round Feistel $F: K^3 \times \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ a secure PRP

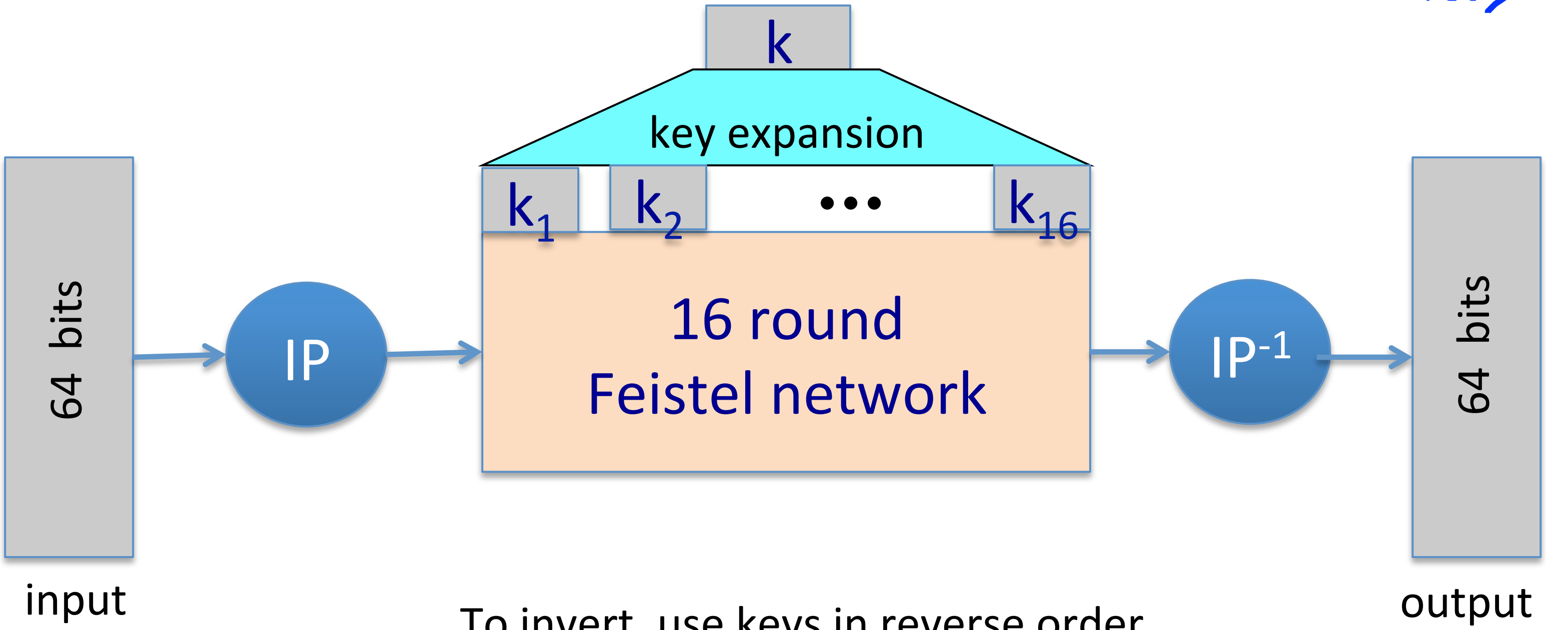


$f(k_0, R_0)$ $f(k_1, R_1)$ $f(k_2, R_2)$ $\Longrightarrow$ 3 independent keys

DES: 16 round Feistel network

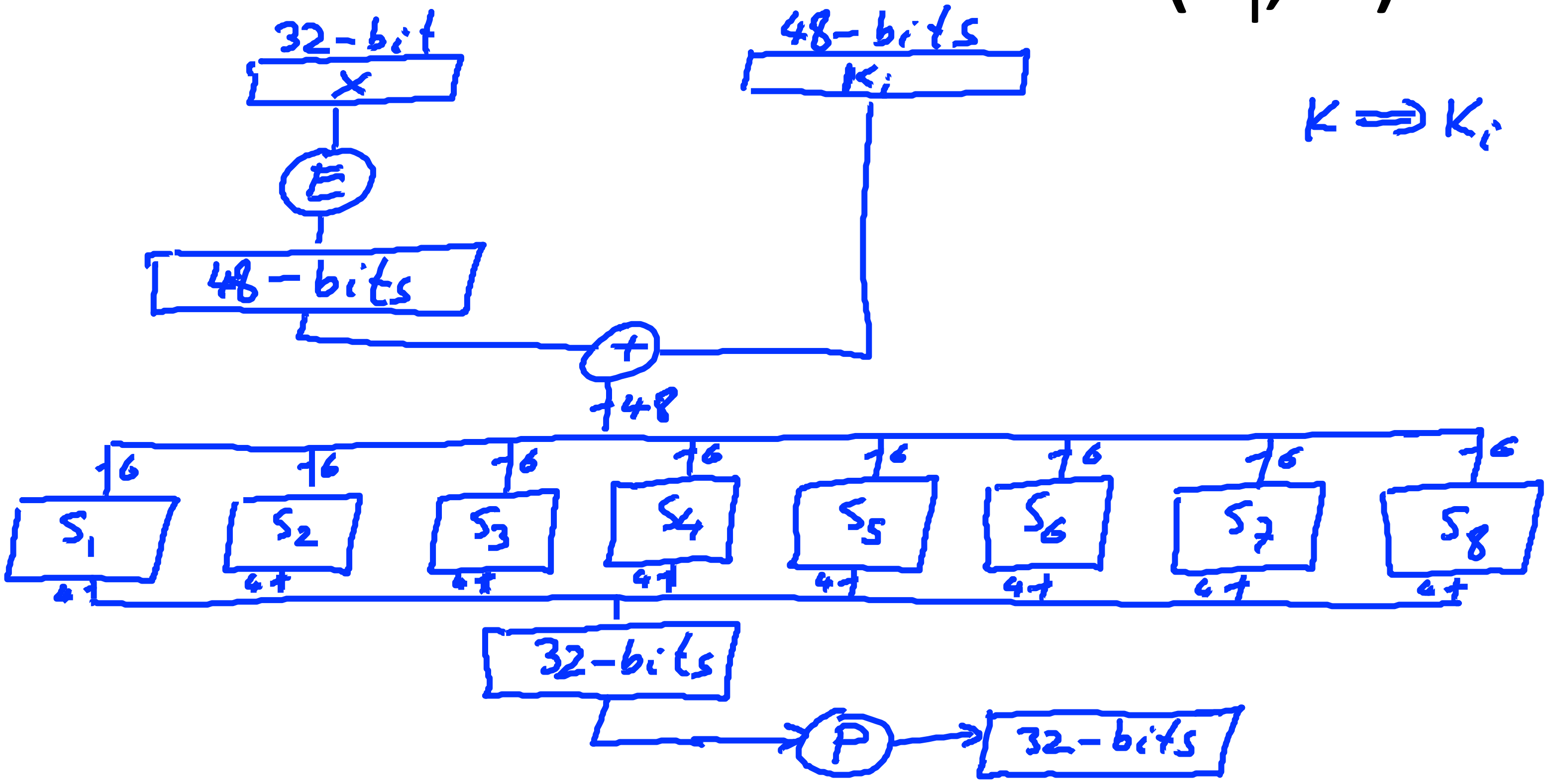
$f_1, \ldots, f_{16}: \{0,1\}^{32} \longrightarrow \{0,1\}^{32}$, $f_i(x) = \mathbf{F}(k_i, x)$

from key k



To invert, use keys in reverse order

The function F(k_i, x)



$K \Rightarrow K_i$

S-box: function $\{0,1\}^6 \longrightarrow \{0,1\}^4$, implemented as look-up table.

The S-boxes

$$S_i: \{0,1\}^6 \longrightarrow \{0,1\}^4$$

S_5		Middle 4 bits of input															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Example: a bad S-box choice

Suppose:

$$S_i(x_1, x_2, \ldots, x_6) = \big(\ x_2 \oplus x_3,\ \ x_1 \oplus x_4 \oplus x_5,\ \ x_1 \oplus x_6,\ \ x_2 \oplus x_3 \oplus x_6\ \big)$$

or written equivalently: $S_i(\mathbf{x}) = A_i \cdot \mathbf{x} \pmod 2$

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} x_2 \oplus x_3 \\ x_1 \oplus x_4 \oplus x_5 \\ x_1 \oplus x_6 \\ x_2 \oplus x_3 \oplus x_6 \end{pmatrix}$$

We say that S_i is a linear function.

Example: a bad S-box choice

Then entire DES cipher would be linear: ∃ fixed binary matrix B s.t.

$$\mathrm{DES}(k,m) = \begin{array}{c} \\ 64 \end{array} \overset{832}{\boxed{B}} \cdot \begin{pmatrix} m \\ k_1 \\ k_2 \\ \vdots \\ k_{16} \end{pmatrix} = \begin{pmatrix} c \end{pmatrix} \pmod 2$$

But then: $\mathrm{DES}(k,m_1) \oplus \mathrm{DES}(k,m_2) \oplus \mathrm{DES}(k,m_3) = \mathrm{DES}(k, m_1 \oplus m_2 \oplus m_3)$

$$k = \begin{pmatrix} k_1 \\ \vdots \\ k_{16} \end{pmatrix}$$

$$B \begin{pmatrix} m_1 \\ k \end{pmatrix} \oplus B \begin{pmatrix} m_2 \\ k \end{pmatrix} \oplus B \begin{pmatrix} m_3 \\ k \end{pmatrix} = B \begin{pmatrix} m_1 \oplus m_2 \oplus m_3 \\ k \oplus k \oplus k \end{pmatrix}$$

Choosing the S-boxes and P-box

Choosing the S-boxes and P-box at random would result in an insecure block cipher (key recovery after $\approx 2^{24}$ outputs) [BS'89]

Several rules used in choice of S and P boxes:

- No output bit should be close to a linear func. of the input bits
- S-boxes are 4-to-1 maps

  ⋮

End of Segment