CS 4770: Cryptography

CS 6750: Cryptography and Communication Security

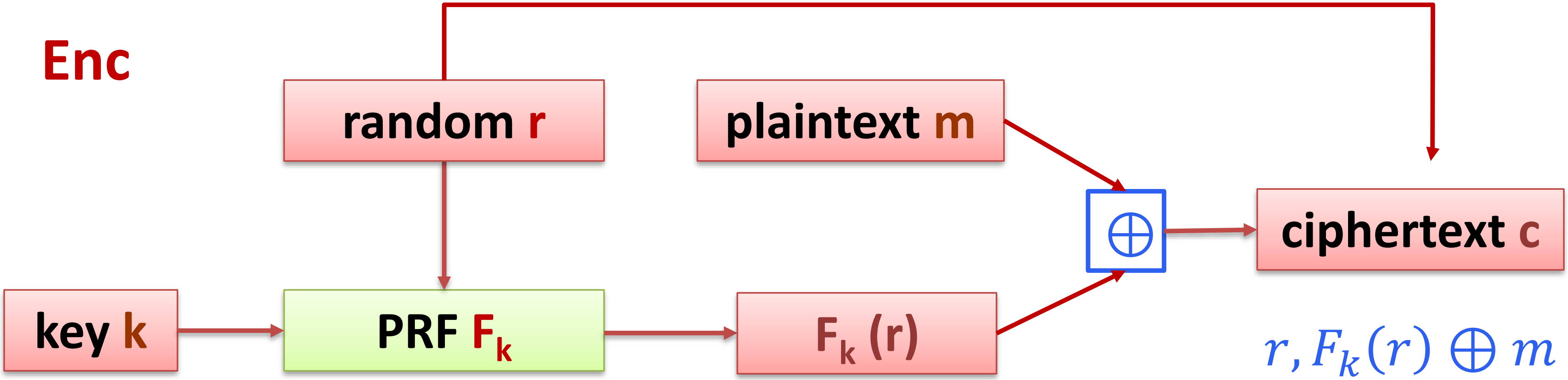
Alina Oprea
Associate Professor, CCIS
Northeastern University

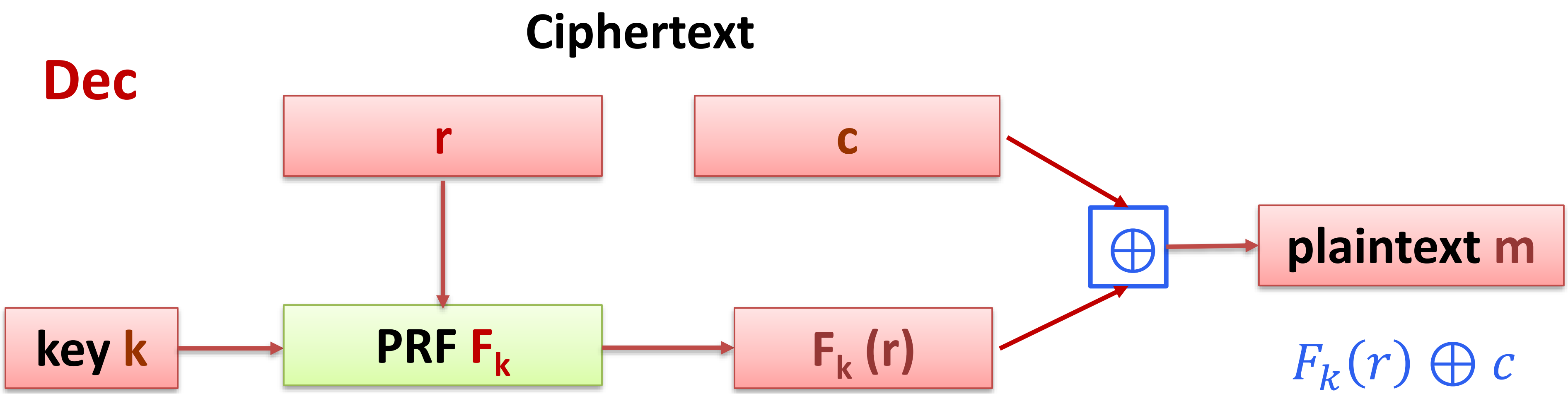
February 5 2018

Review

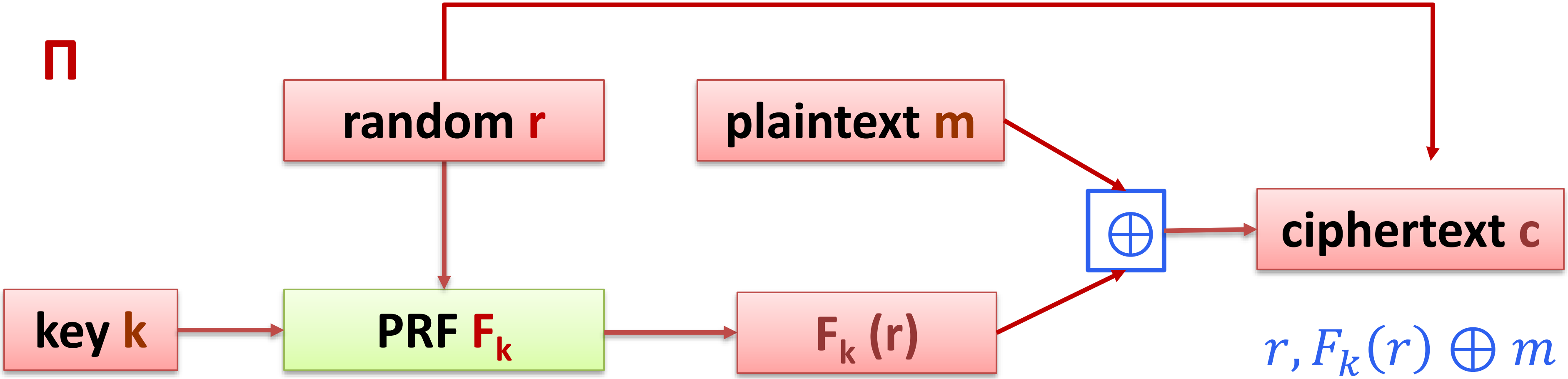
- Relation between PRF and PRG
  - Construct PRF from PRG (GGM construction)
- Pseudorandom permutations
- Definitions of security for encryption
  - CPA/CCA security
  - Relations between definitions
- CPA-secure construction
  - Security proof
  - Reduction to PRF

How to encrypt using PRF?

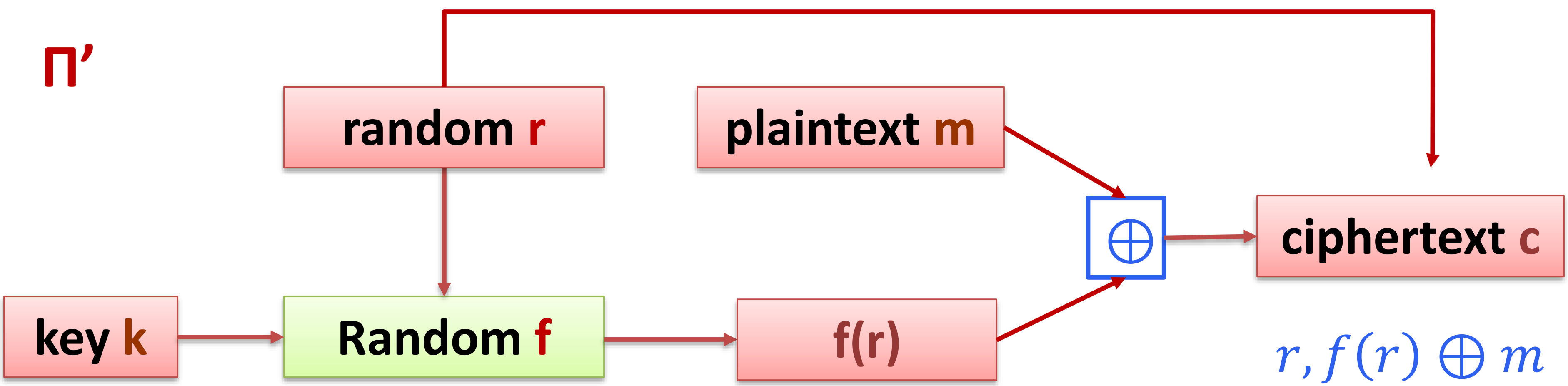




Proof of security - Intuition



$r, F_k(r) \oplus m$



$r, f(r) \oplus m$

Proof of security - Intuition

Π

Enc

$$c = (r, F_k(r) \oplus m)$$

Dec

$$c = (r, s)$$

$$m = F_k(r) \oplus s$$

1. Success of adversary to break **Π** and **Π'** in CPA game is similar

Under the assumption that F is a PRF!

Π'

Enc

$$c = (r, f(r) \oplus m)$$

Dec

$$c = (r, s)$$

$$m = f(r) \oplus s$$

2. Success of adversary to break **Π'** in CPA game is negligible

Proof of security – step 2

2. Success of adversary to break **Π'** in CPA game is negligible

For any adversary A that makes q(n) queries to Enc oracle:

$$\Pr[\mathrm{Exp}^{\mathrm{CPA}}_{\Pi',A}(n) = 1] - \frac{1}{2} \text{ is } negl(n)$$

- Let A be an adversary in CPA game for Π' that makes q = q(n) queries
- For each query to Enc oracle $m_1, \cdots, m_q$, it gets back $c_i = (r_i, f(r_i) \oplus m_i)$
- A picks m_0, m_1 and receives back $c = (r, f(r) \oplus m_b)$

Proof of security – step 2

2. Success of adversary to break **Π'** in CPA game is negligible

For any adversary A that makes q(n) queries to Enc oracle:

$$\mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = 1] - \frac{1}{2} \; \boldsymbol{is\ negl(n)}$$

- Case 1 - r is not used to answer the q queries to Enc : $\mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = 1] = \frac{1}{2}$
- Case 2 - $r \in \{r_1, \cdots, r_q\}$: $\mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = 1] = 1$
  – But $\Pr[r \in \{r_1, \cdots, r_q\}] \leq \sum_i \Pr[r = r_i] \quad \leq q(n)/2^n$

$$\mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = 1] \leq \frac{1}{2} + \frac{q(n)}{2^n}$$

Wrap up

1. Success of adversary to break **Π** and **Π'** in CPA game is similar

Assume that F is secure PRF.

For any adversary A that makes q(n) queries to Enc oracle:

$$\left|\mathbf{Pr}[\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n) = \mathbf{1}] - \mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = \mathbf{1}]\right| \leq \mathrm{negl(n)}$$

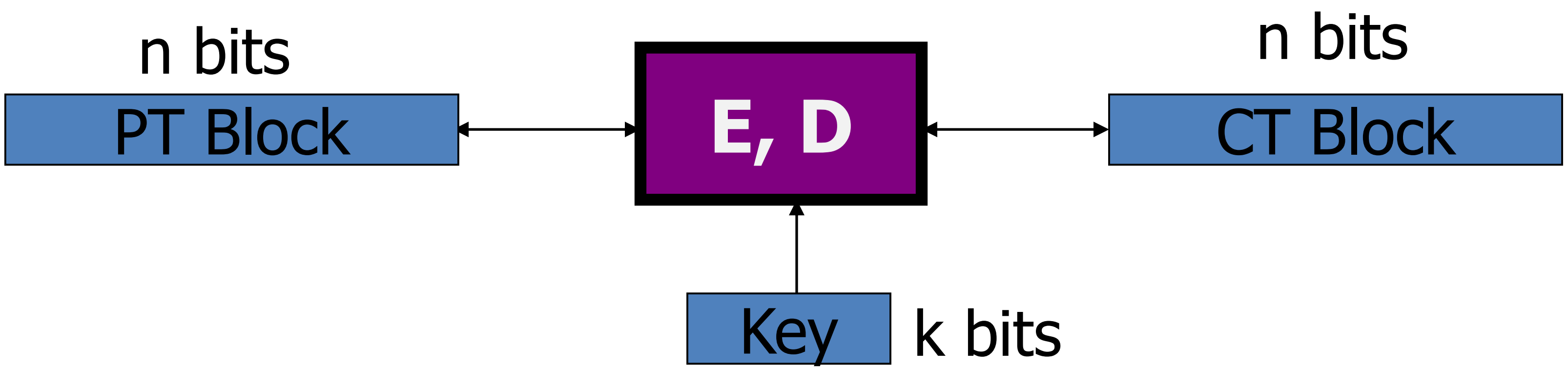
2. Success of adversary to break **Π'** in CPA game is negligible

For any adversary A that makes q(n) queries to Enc oracle:

$$\mathbf{Pr}[\mathrm{Exp}_{\Pi',A}^{\mathrm{CPA}}(n) = \mathbf{1}] \leq \frac{\mathbf{1}}{\mathbf{2}} + \frac{\boldsymbol{q(n)}}{\mathbf{2}^{\boldsymbol{n}}}$$

$$\mathbf{Pr}[\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n) = \mathbf{1}] \leq \frac{\mathbf{1}}{\mathbf{2}} + \frac{\boldsymbol{q(n)}}{\mathbf{2}^{\boldsymbol{n}}} + \mathrm{negl(n)}$$

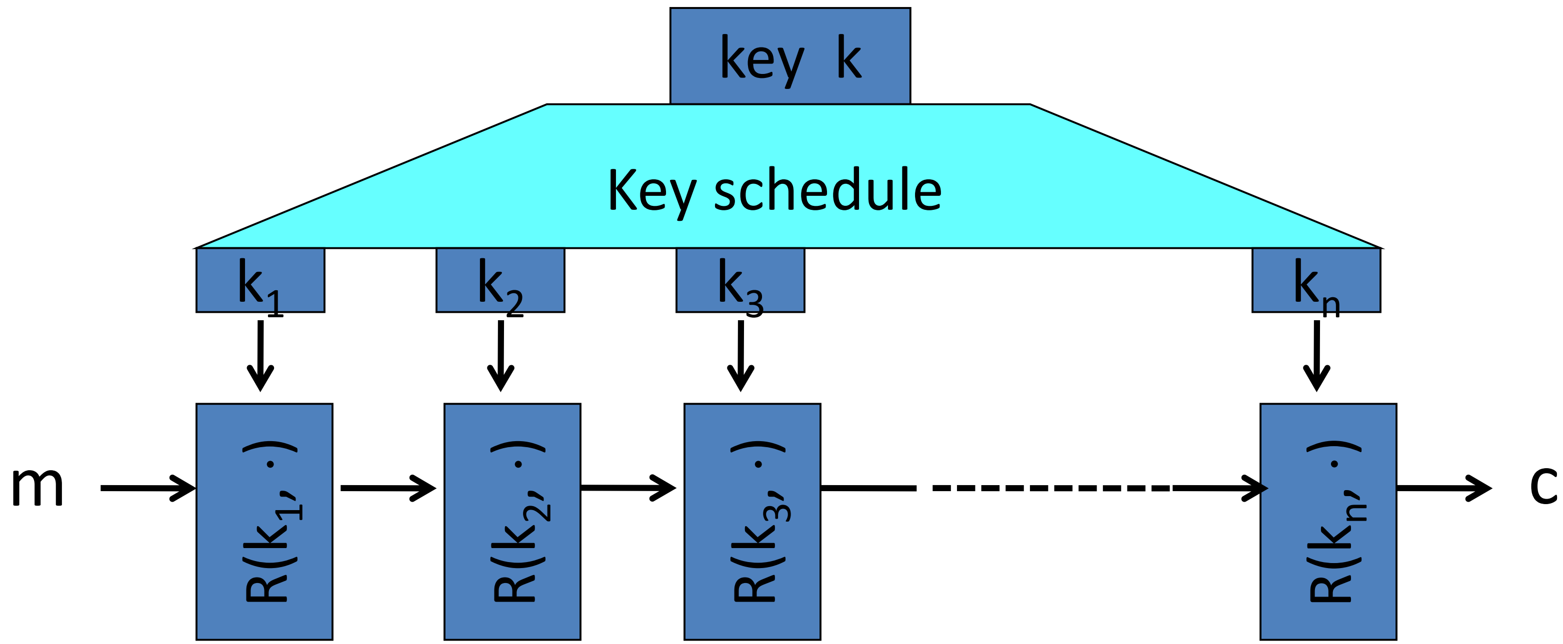
Block ciphers: crypto work horse



Canonical examples:

1. DES: n= 64 bits, k = 56 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 DES (n=48), for AES-128 (n=10)

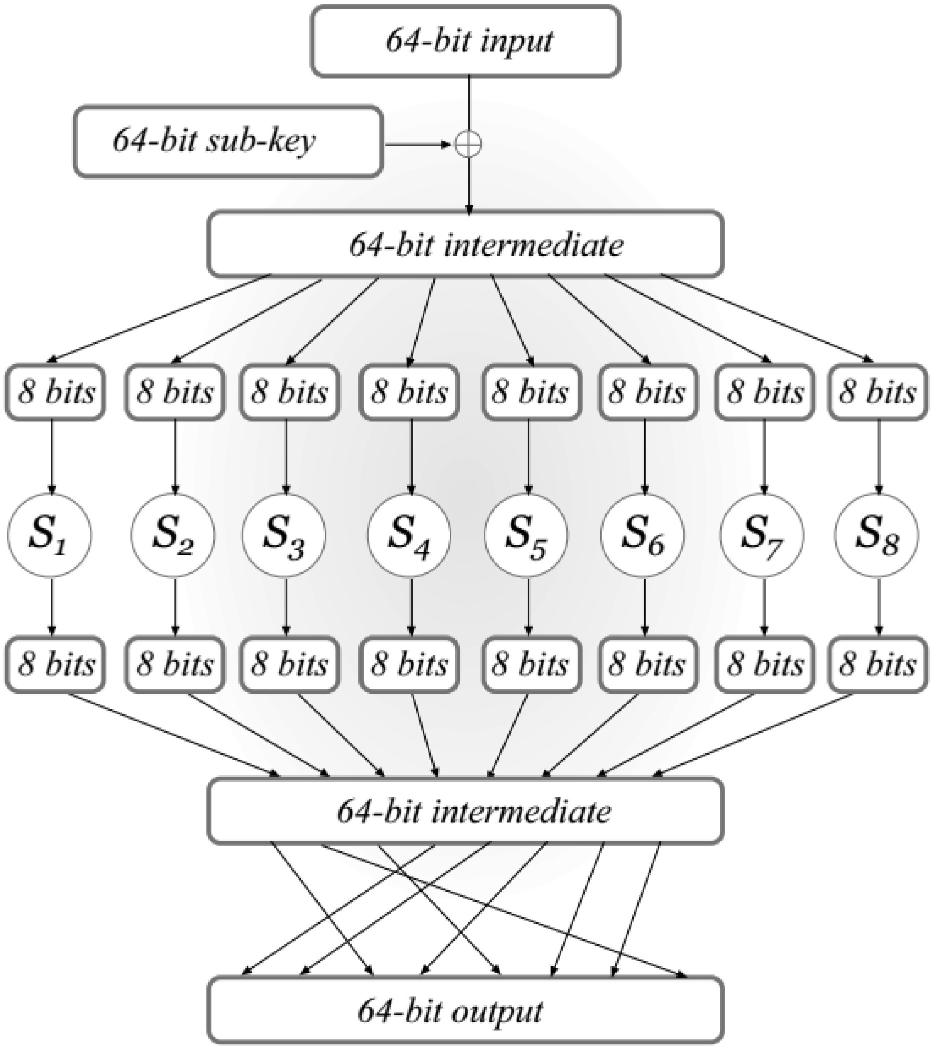
Design goals

- Block ciphers should behave like random permutations
  - The number of permutation for n-bit strings is $(2^n)! \approx n2^n$
  - Construct set of permutations with concise description (short key)
  - Similar to security property of PRP
- Properties
  - Changing one bit of input should affect all bits of output (good mixing)
- Two main design approaches
  - Substitution-Permutation Network
  - Feistel Network

Substitution-Permutation Network

Round key

S-box
Fixed permutation
Invertible



Key mixing

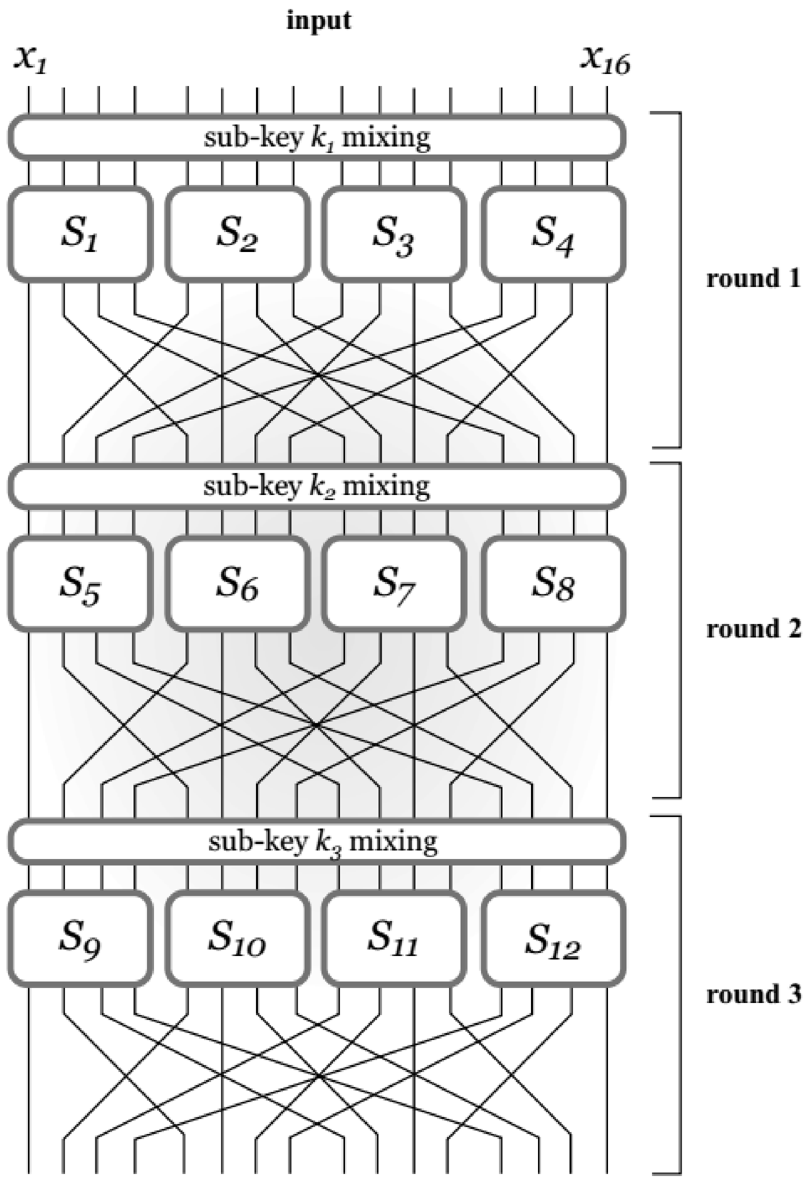
Substitution

Permutation

S boxes and mixing permutation are public

Three rounds of SPN

Invertible if key known



1. Key mixing
2. S boxes
3. Mixing permutation
4. Number of rounds

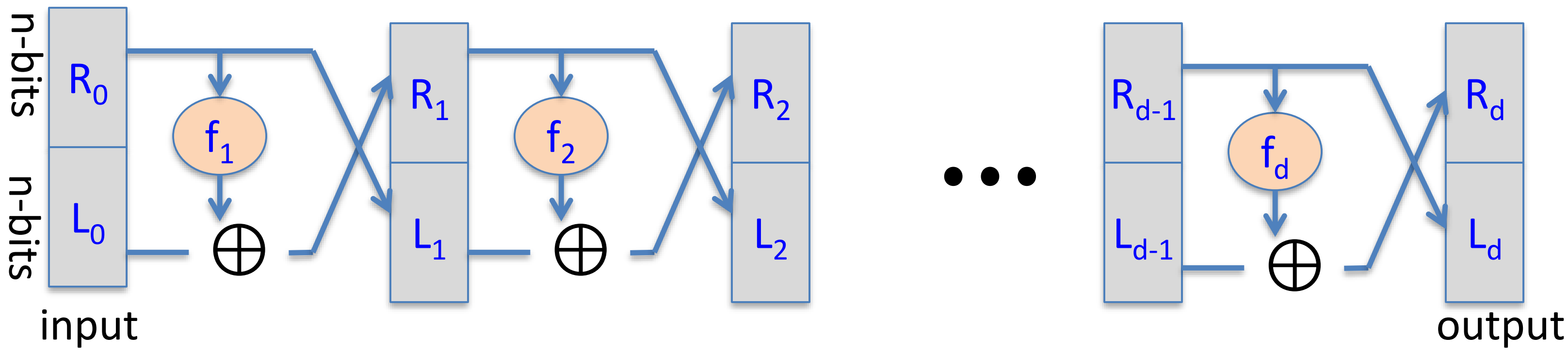
The avalanche effect

- Changing *a single bit of input* in S box changes *at least 2 bits of output* in S box
- The mixing permutations ensure that the *output bits of any S box* are used as *input to multiple S boxes* in the next round

Feistel Networks

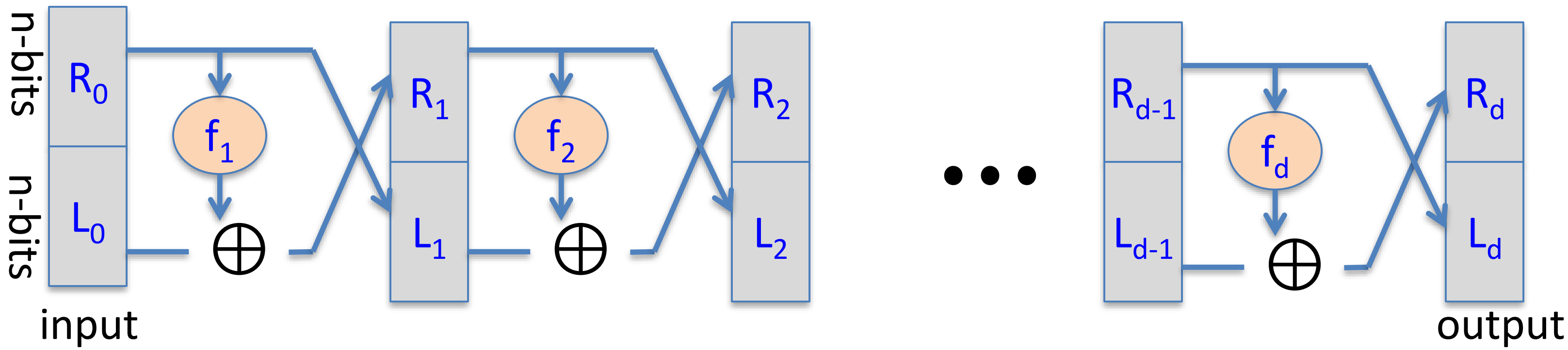
Given functions $f_1, \ldots, f_d$: $\{0,1\}^n \longrightarrow \{0,1\}^n$

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



$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f_i(R_{i-1})$$

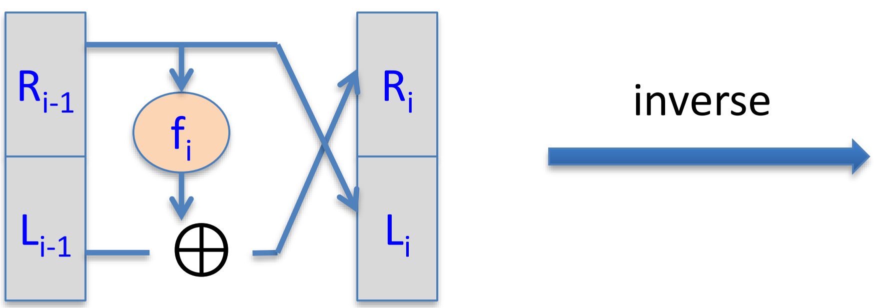
- Functions f_i are public
- Round key is derived from main key and secret
- Advantage: f_i not invertible!



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

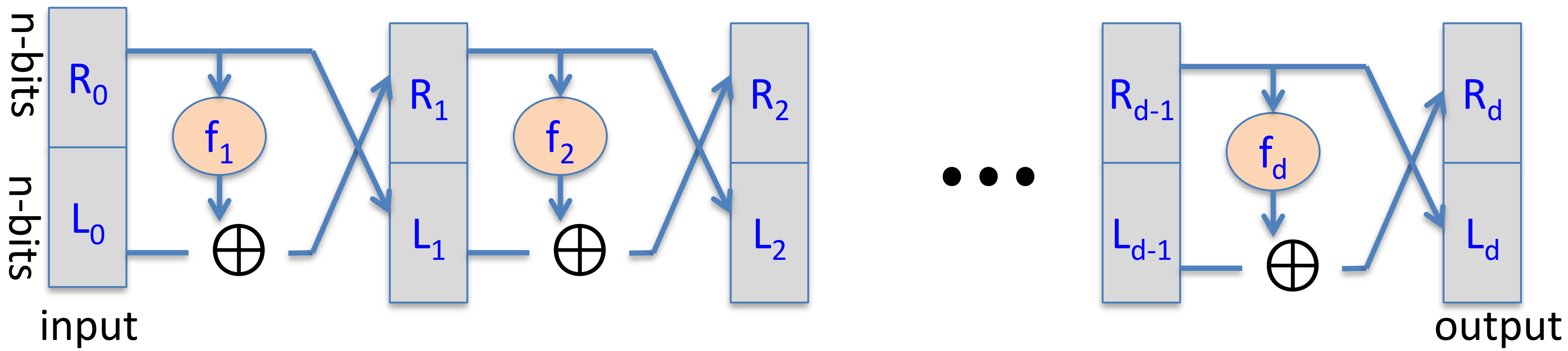
Feistel network $F: \{0,1\}^{2n} \longrightarrow \{0,1\}^{2n}$ is invertible

Proof: construct inverse



$R_{i-1} = L_i$

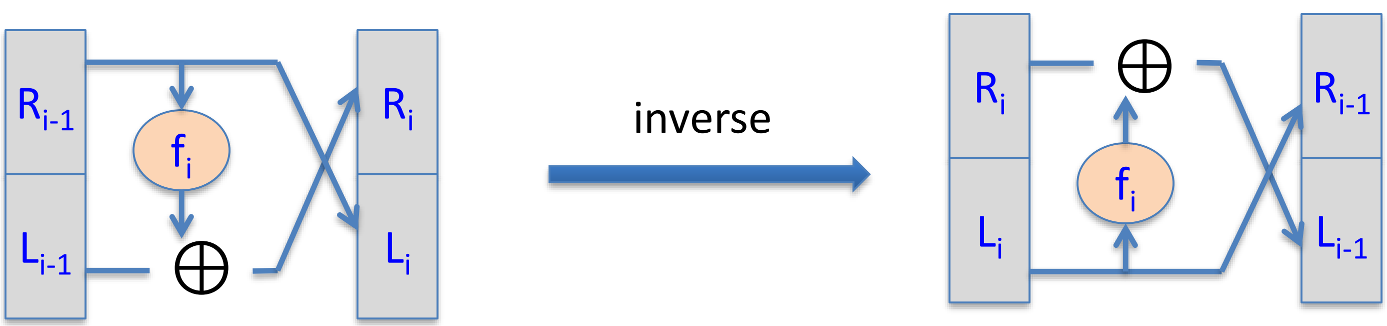
$L_{i-1} =$



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

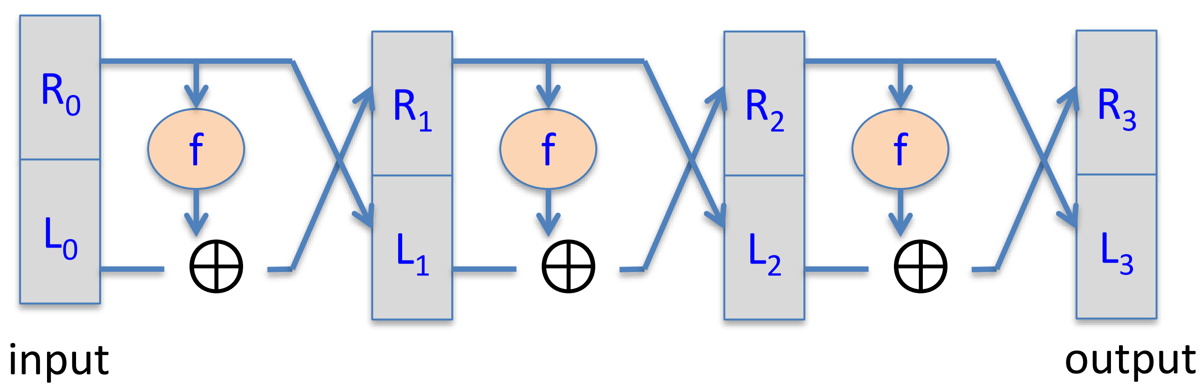
Proof: construct inverse



“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



Key k_1 Key k_2 Key k_3 Independent

The Data Encryption Standard (DES)

- Early 1970s: Horst Feistel designs Lucifer at IBM

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

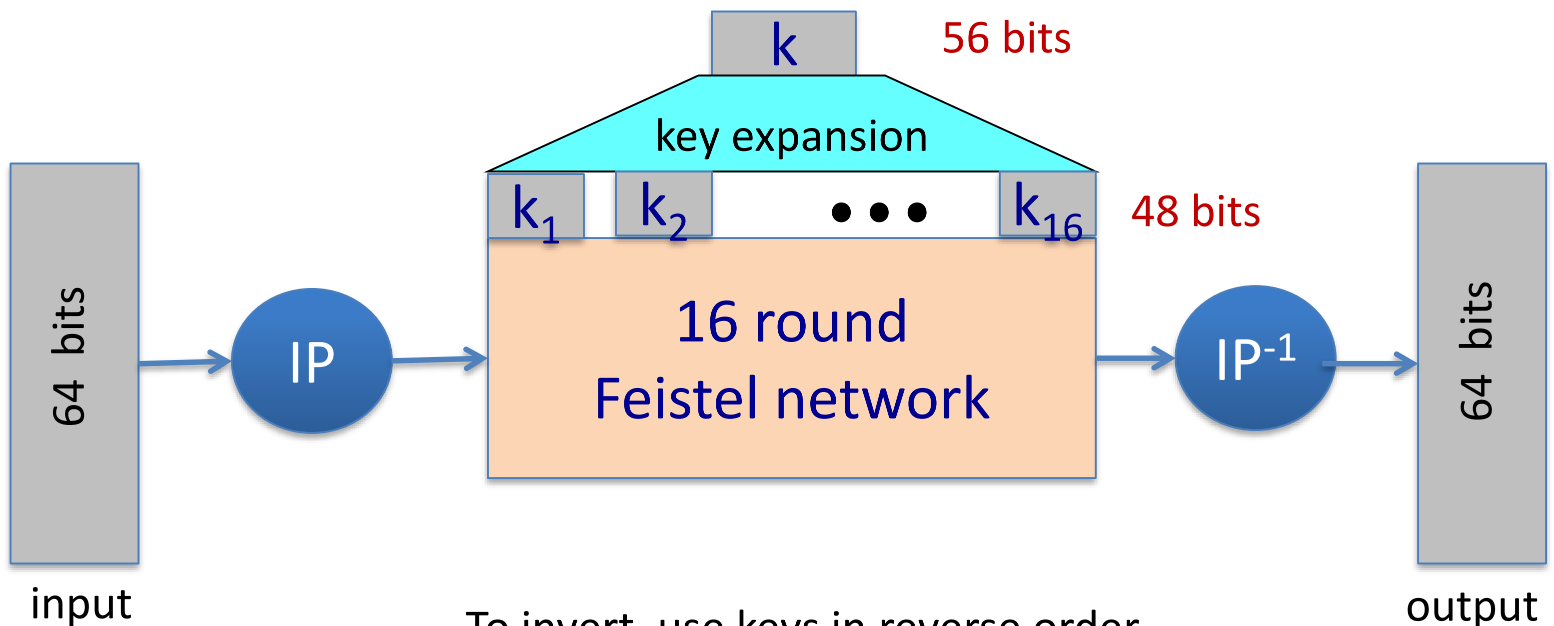
- 1973: NBS asks for block cipher proposals.
  IBM submits variant of Lucifer.
- 1976: NBS adopts DES as a federal standard

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

- 1997: DES broken by exhaustive search
- 2000: NIST adopts Rijndael as AES to replace DES

DES: 16 round Feistel network

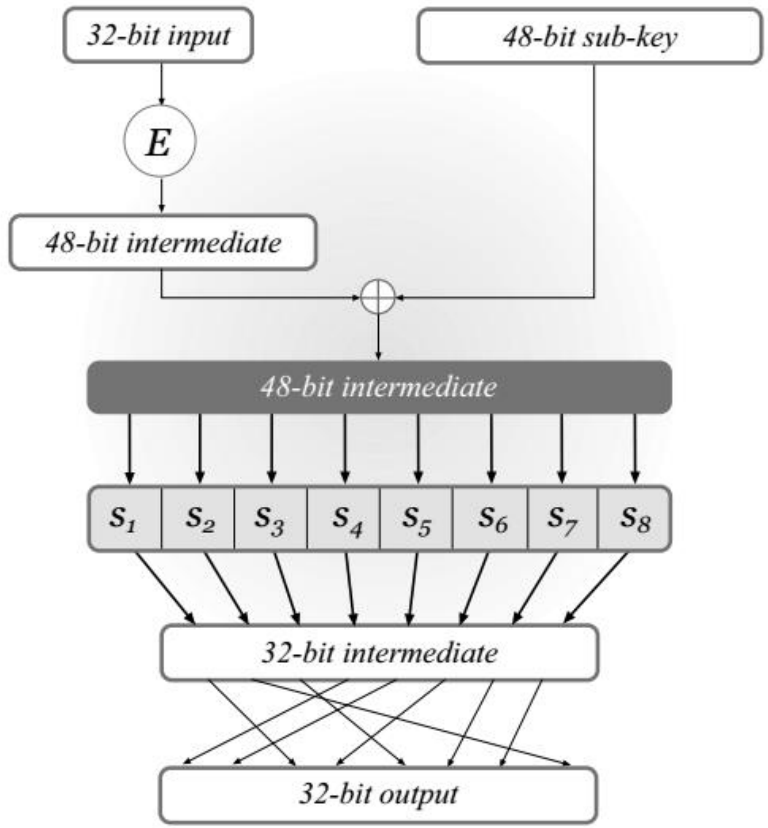
$f_1, \ldots, f_{16}: \{0,1\}^{32} \longrightarrow \{0,1\}^{32}$, $f_i(x) = \mathbf{F}(k_i, x)$



To invert, use keys in reverse order

The function $F(k_i, x)$

Substitution-Permutation Network



Key mixing

Substitution

Permutation

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

The S-boxes

Look up table

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

$x_2 x_3 x_4 x_5$

$x_1 x_6$

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

$x_1 x_2 x_3 x_4 x_5 x_6$

Not invertible

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 ≈2^{24} outputs) [BS'89]

Several rules used in choice of S and P boxes:

- No output bit should be close to a linear function of the input bits
- S-boxes are 4-to-1 maps (Exactly 4 inputs are mapped to each output)
- Each row in the table contains each 4-bit string exactly once
- Changing one bit of input to S box results in changing 2 bits of output

DES challenge

msg = `"The unknown messages is: XXXX … "`

CT = c_1 c_2 c_3 c_4

Goal: find $k \in \{0,1\}^{56}$ s.t. $DES(k, m_i) = c_i$ for $i=1,2,3$

1997: Internet search -- **3 months**

1998: EFF machine (deep crack) -- **3 days** (250K $)

1999: combined search -- **22 hours**

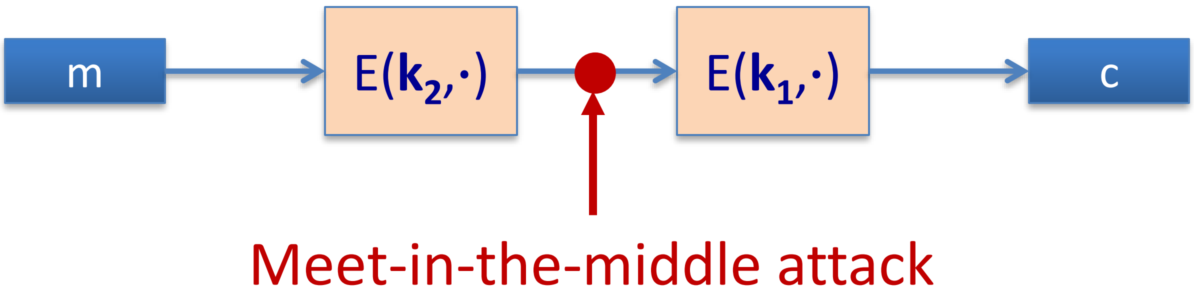
2006: COPACOBANA (120 FPGAs) **-- 7 days** (10K $)

⇒ 56-bit ciphers should not be used !! (128-bit key ⇒ 2^{72} days)

Double DES

- Define $2E((k_1,k_2), m) = E(k_1, E(k_2, m))$

key length = 112 bits for DES

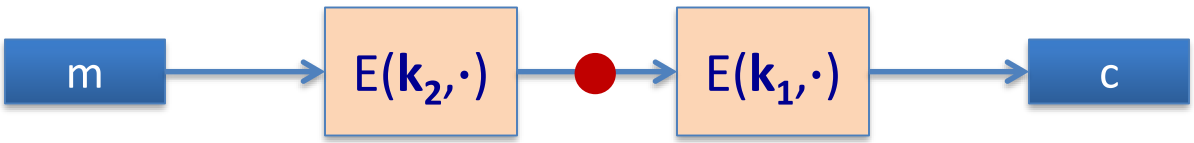


- Find (k_1, k_2) such that $E(k_1, E(k_2, m)) = C$
- Equivalent to $E(k_2, m) = D(k_1, m)$

Double DES

- Define $2E(\,(k_1,k_2),\, m) = E(k_1\,,\, E(k_2\,,\, m)\,)$

key-len = 112 bits for DES



Attack: $M = (m_1,\ldots,m_u)\ ,\ \ C = (c_1,\ldots,c_u)$

- step 1: build table.

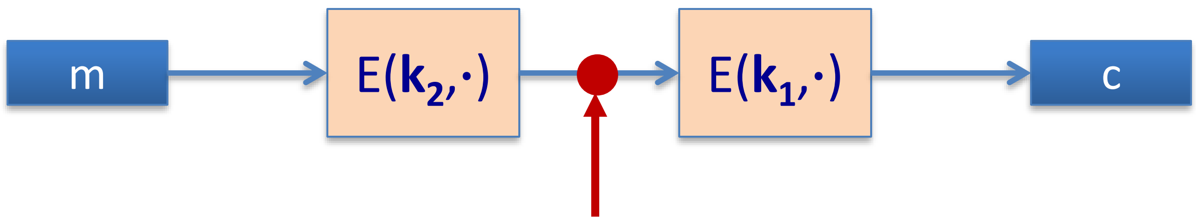
  sort on 2^{nd} column

$k^0 = 00\ldots00$	$E(k^0\,,\, M)$
$k^1 = 00\ldots01$	$E(k^1\,,\, M)$
$k^2 = 00\ldots10$	$E(k^2\,,\, M)$
⋮	⋮
$k^N = 11\ldots11$	$E(k^N\,,\, M)$

2^{56} entries

Time $2^{56}\log(2^{56})$

Meet in the middle attack



Attack: $M = (m_1,\ldots, m_u)$, $C = (c_1,\ldots,c_u)$

- Step 1: build table.

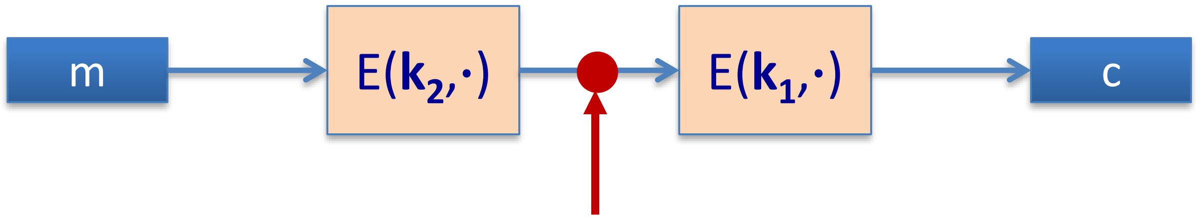
$k^0 = 00\ldots00$	$E(k^0 , M)$
$k^1 = 00\ldots01$	$E(k^1 , M)$
$k^2 = 00\ldots10$	$E(k^2 , M)$
⋮	⋮
$k^N = 11\ldots11$	$E(k^N , M)$

- Step 2: for all $k\in\{0,1\}^{56}$ do:

  test if $D(k, C)$ is in 2nd column.

if so then $E(k^i,M) = D(k,C) \Rightarrow (k^i,k) = (k_2,k_1)$

Meet in the middle attack



Time = $2^{56}\log(2^{56}) + 2^{56}\log(2^{56}) < 2^{63} \ll 2^{112}$

Build table — Search table

Space ≈ 2^{56}

Triple DES

- Let $E : K \times M \longrightarrow M$ be a block cipher
- Define **3E**: $K^3 \times M \longrightarrow M$ as

$$3E(\ (k_1,k_2,\ k_3),\ m) = \ E(k_1\ ,\ D(k_2\ ,E(k_3\ ,\ m)\ )\ )$$

If $k_1 = k_2 = k_3$ then 3E = DES!

For 3DES: key-size = 3×56 = 168 bits

3×slower than DES

(simple attack in time $\approx 2^{118}$)

The AES process

- 1997: NIST publishes request for proposal
- 1998: 15 submissions. Five claimed attacks.
- 1999: NIST chooses 5 finalists
- 2000: NIST chooses Rijndael as AES (designed in Belgium)

Key sizes: 128, 192, 256 bits.

Block size: 128 bits

Acknowledgement

Some of the slides and slide contents are taken from
http://www.crypto.edu.pl/Dziembowski/teaching
and fall under the following:

We have also used slides from Prof. Dan Boneh online cryptography course at Stanford University:

http://crypto.stanford.edu/~dabo/courses/OnlineCrypto/