CSL759: Cryptography and Computer Security

Ragesh Jaiswal
CSE, IIT Delhi

Block Ciphers

Block Ciphers: Introduction

- Block ciphers work on "blocks" of message bits rather than a "stream" of message bits.
- Main Idea:
  - Suppose we encrypt in blocks of size n.
  - Let $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a function.
  - For a message block M of n bits, and key K, the ciphertext is given by $C = E(K, M)$.

Block Ciphers: Introduction

- Block ciphers work on "blocks" of message bits rather than a "stream" of message bits.
- Main Idea:
  - Suppose we encrypt in blocks of size n.
  - Let $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a function.
  - For a message block M of n bits, and key K, the ciphertext is given by $C = E(K, M)$.
- What are properties that E should satisfy?

Block Ciphers: Introduction

- Block ciphers work on "blocks" of message bits rather than a "stream" of message bits.
- Main Idea:
  - Suppose we encrypt in blocks of size n.
  - Let $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a function.
  - For a message block M of n bits, and key K, the ciphertext is given by $C = E(K, M)$.
- What are properties that E should satisfy?
  - For all $K \in \{0,1\}^k$, the function $\mathrm{E_K}: \{0,1\}^{\mathrm{n}} \rightarrow \{0,1\}^n$ defined as $E_K(M) = E(K, M)$ is a one-one function. In other words, E_K is a permutation.

Block Ciphers: Introduction

- Block ciphers work on "blocks" of message bits rather than a "stream" of message bits.
- Main Idea:
  - Suppose we encrypt in blocks of size n.
  - Let $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a function.
  - For a message block M of n bits, and key K, the ciphertext is given by $C = E(K, M)$.
- What are properties that E should satisfy?
  - For all $K \in \{0,1\}^k$, the function $\mathrm{E_K}: \{0,1\}^{\mathrm{n}} \rightarrow \{0,1\}^n$ defined as $E_K(M) = E(K, M)$ is a one-one function. In other words, E_K is a permutation.
  - Both E_K (encryption function) and E_K^{-1} (decryption function) are efficient.
  - **Security Properties**: To be discussed.

Block Ciphers: Introduction

- Block ciphers work on "blocks" of message bits rather than a "stream" of message bits.
- Main Idea:
  - Suppose we encrypt in blocks of size n.
  - Let $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a function.
  - For a message block M of n bits, and key K, the ciphertext is given by $C = E(K, M)$.
- What are properties that E should satisfy?
  - For all $K \in \{0,1\}^k$, the function $\mathrm{E_K}: \{0,1\}^{\mathrm{n}} \rightarrow \{0,1\}^n$ defined as $E_K(M) = E(K, M)$ is a one-one function. In other words, E_K is a permutation.
  - Both E_K (encryption function) and E_K^{-1} (decryption function) are efficient.
  - **Security Properties**: To be discussed.

M

$C = E_K(M)$

$M = D_K(C) = E_K^{-1}(C)$

K ⟷ K

Key exchange protocol

Block Ciphers: Introduction

- Block ciphers: Examples:
  - DES: $\{0,1\}^{56} \times \{0,1\}^{64} \to \{0,1\}^{64}$
  - 3DES: $\{0,1\}^{168} \times \{0,1\}^{64} \to \{0,1\}^{64}$
  - AES: $\{0,1\}^{k} \times \{0,1\}^{128} \to \{0,1\}^{128}, k = 128, 192, 256.$

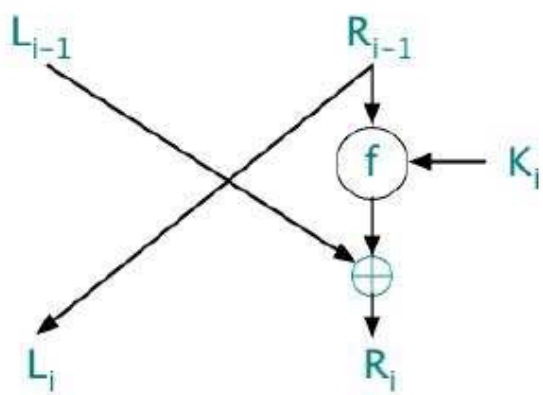
- Data Encryption Standard (DES):
  - Early 1970's: Horst Feistel designs a block cipher *Lucifer* at IBM.
  - 1973: NBS (now NIST) asks for a block cipher for standardization. IBM submits a variant of *Lucifer*.
  - 1976: NBS adopts DES as a Federal standard.
  - 1997: DES broken by exhaustive search.
  - 2000: NIST adopts *Rijndael* as AES to replace DES.

Block Ciphers: DES

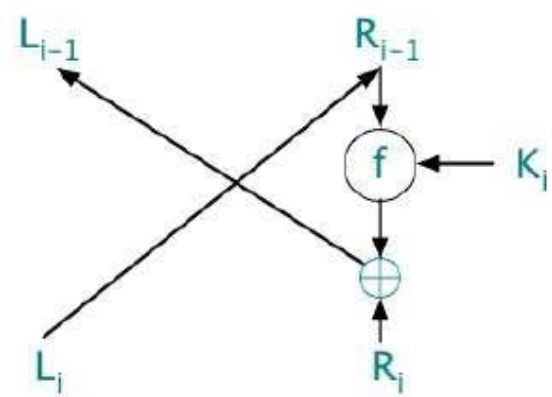
DES Construction

function $\mathrm{DES}_K(M)$ // $|K| = 56$ and $|M| = 64$
  $(K_1, \ldots, K_{16}) \leftarrow KeySchedule(K)$ // $|K_i| = 48$ for $1 \le i \le 16$
  $M \leftarrow IP(M)$
  Parse M as $L_0 \parallel R_0$ // $|L_0| = |R_0| = 32$
  for $i = 1$ to 16 **do**
    $L_i \leftarrow R_{i-1}$; $R_i \leftarrow f(K_i, R_{i-1}) \oplus L_{i-1}$
  $C \leftarrow IP^{-1}(L_{16} \parallel R_{16})$
  return C

Round i:

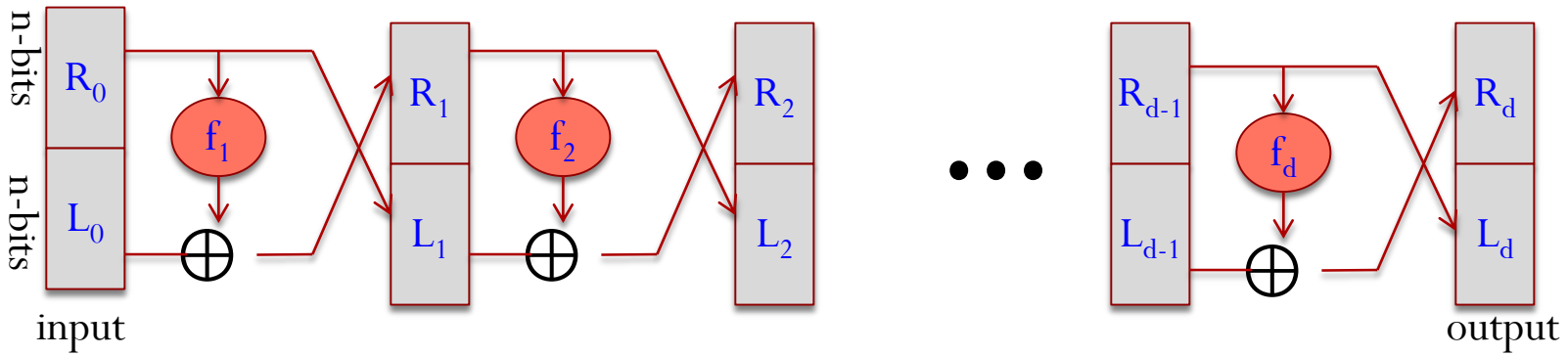


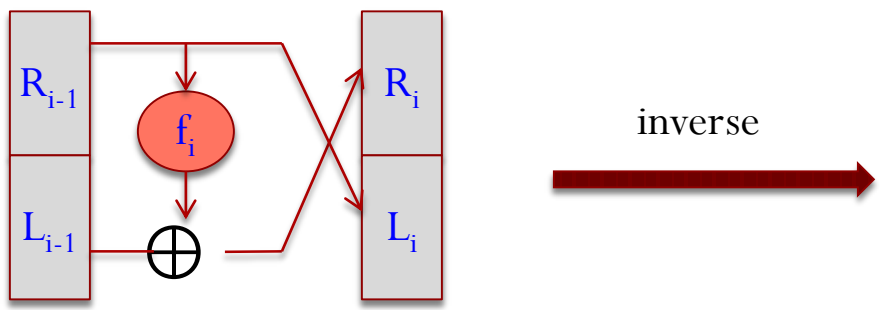
Invertible given K_i:



Block Ciphers: DES

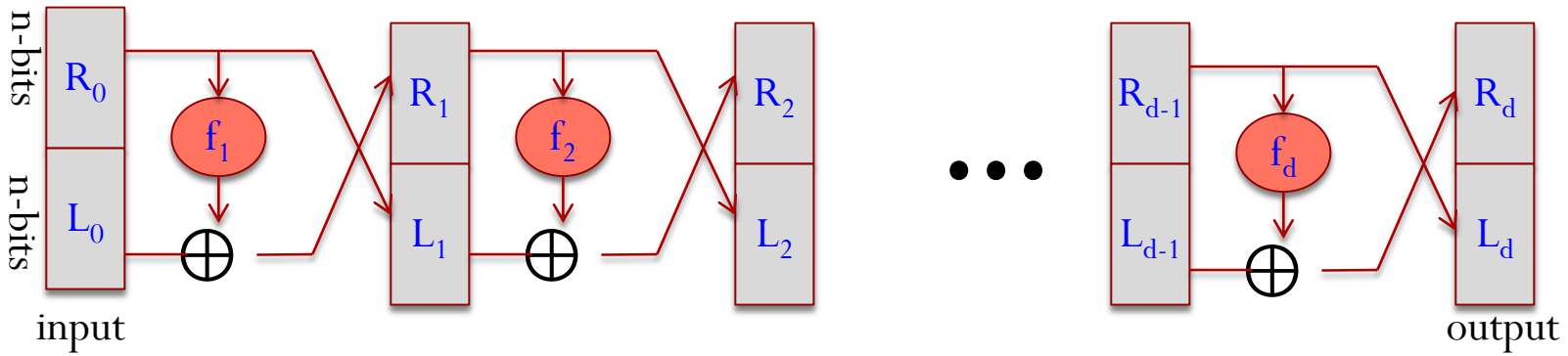
Feistel Network

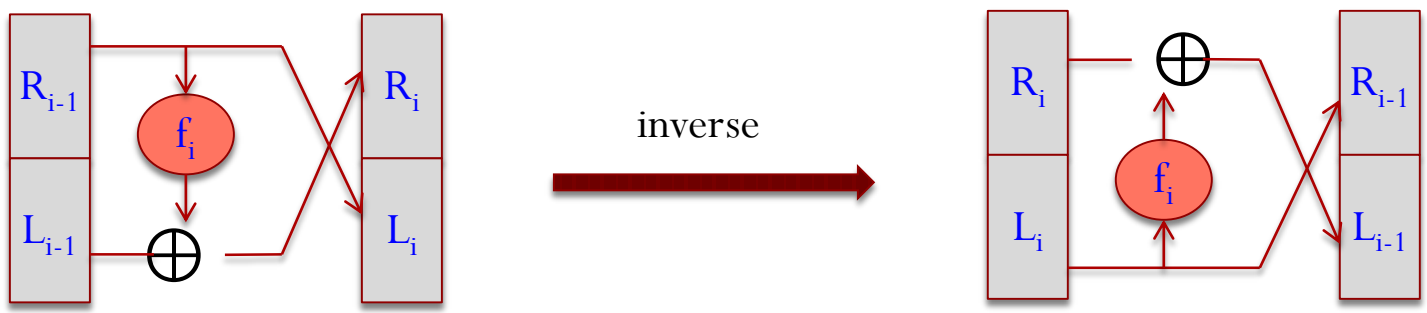




Block Ciphers: DES

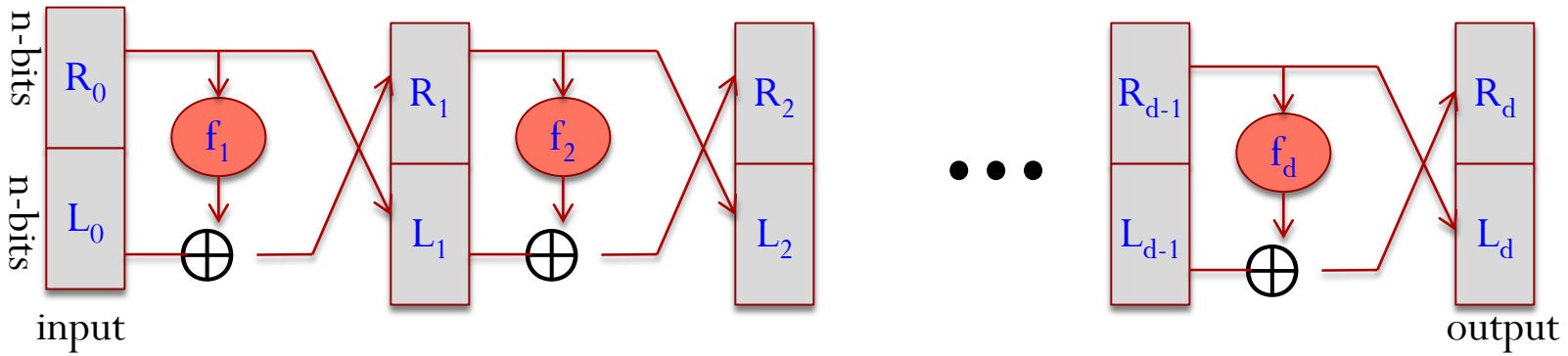
Feistel Network



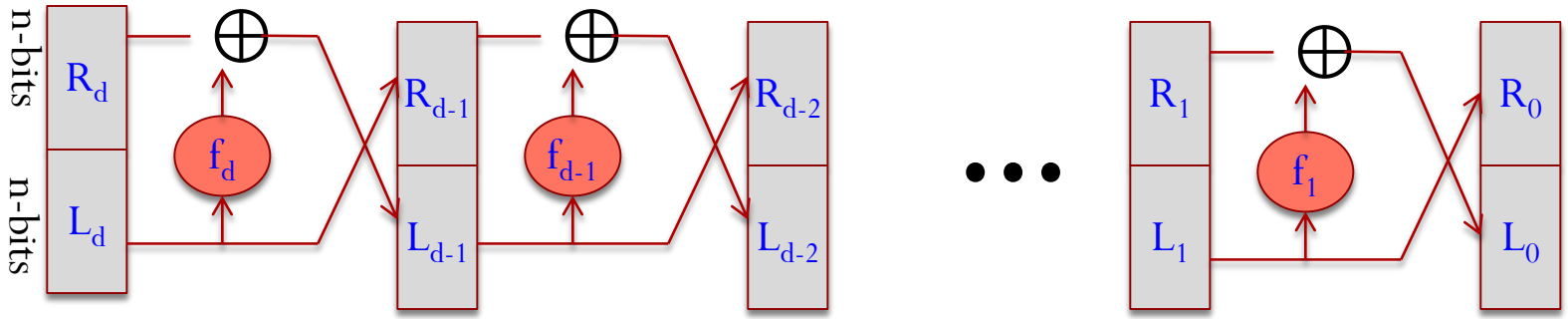


Block Ciphers: DES

Encryption circuit



Decryption circuit



Block Ciphers: DES

DES Construction

function $\mathrm{DES}_K(M)$ // $|K| = 56$ and $|M| = 64$
  $(K_1, \ldots, K_{16}) \leftarrow KeySchedule(K)$ // $|K_i| = 48$ for $1 \leq i \leq 16$
  $M \leftarrow IP(M)$
  Parse M as $L_0 \parallel R_0$ // $|L_0| = |R_0| = 32$
  for $i = 1$ to 16 **do**
    $L_i \leftarrow R_{i-1}$; $R_i \leftarrow f(K_i, R_{i-1}) \oplus L_{i-1}$
  $C \leftarrow IP^{-1}(L_{16} \parallel R_{16})$
  return C

function $\mathrm{DES}_K^{-1}(C)$ // $|K| = 56$ and $|M| = 64$
  $(K_1, \ldots, K_{16}) \leftarrow KeySchedule(K)$ // $|K_i| = 48$ for $1 \leq i \leq 16$
  $C \leftarrow IP(C)$
  Parse C as $L_{16} \parallel R_{16}$
  for $i = 16$ downto 1 **do**
    $R_{i-1} \leftarrow L_i$; $L_{i-1} \leftarrow f(K_i, R_{i-1}) \oplus R_i$
  $M \leftarrow IP^{-1}(L_0 \parallel R_0)$
  return M

Block Ciphers: DES

DES Construction

function $\mathrm{DES}_K(M)$ // $|K| = 56$ and $|M| = 64$
  $(K_1, \ldots, K_{16}) \leftarrow KeySchedule(K)$ // $|K_i| = 48$ for $1 \leq i \leq 16$
  $M \leftarrow IP(M)$
  Parse M as $L_0 \parallel R_0$ // $|L_0| = |R_0| = 32$
  for $i = 1$ to 16 **do**
    $L_i \leftarrow R_{i-1}$; $R_i \leftarrow f(K_i, R_{i-1}) \oplus L_{i-1}$
  $C \leftarrow IP^{-1}(L_{16} \parallel R_{16})$
  return C

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Block Ciphers: DES

DES Construction

function $f(J, R)$ // $|J| = 48$ and $|R| = 32$

$R \leftarrow E(R)$; $R \leftarrow R \oplus J$

Parse R as $R_1 \parallel R_2 \parallel R_3 \parallel R_4 \parallel R_5 \parallel R_6 \parallel R_7 \parallel R_8$ // $|R_i| = 6$ for $1 \leq i$

for $i = 1, \ldots, 8$ **do**

$R_i \leftarrow \mathbf{S}_i(R_i)$ // Each S-box returns 4 bits

$R \leftarrow R_1 \parallel R_2 \parallel R_3 \parallel R_4 \parallel R_5 \parallel R_6 \parallel R_7 \parallel R_8$ // $|R| = 32$ bits

$R \leftarrow P(R)$

return R

E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Block Ciphers: DES

- The S boxes map $\{0,1\}^6$ to $\{0,1\}^4$

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

- How do we choose S boxes?

Block Ciphers: DES

- The S boxes map $\{0,1\}^6$ to $\{0,1\}^4$

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

- How do we choose S boxes?
- Supose we use S boxes of the following kind:
  - $S_i(x_1, x_2, \ldots, x_6) = (x_2 \oplus x_3, x_1 \oplus x_4 \oplus x_5, x_1 \oplus x_6, x_2 \oplus x_3 \oplus x_6)$
- Do you see a problem using such S boxes?

Block Ciphers: DES

- The S boxes map $\{0,1\}^6$ to $\{0,1\}^4$

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

- How do we choose S boxes?
- Supose we use S boxes of the following kind:
  - $S_i(x_1, x_2, \ldots, x_6) = (x_2 \oplus x_3, x_1 \oplus x_4 \oplus x_5, x_1 \oplus x_6, x_2 \oplus x_3 \oplus x_6)$
- Do you see a problem using such S boxes?
  - The cipher would be linear.

Block Ciphers: DES

- How do we choose S boxes?
- Supose we use S boxes of the following kind:
  - $S_i(x_1, x_2, \ldots, x_6) = (x_2 \oplus x_3, x_1 \oplus x_4 \oplus x_5, x_1 \oplus x_6, x_2 \oplus x_3 \oplus x_6)$
- Do you see a problem using such S boxes?
  - The cipher would be linear.

$$DES(K, m) = \overset{832}{\underset{64}{\boxed{\mathrm{B}}}} \cdot \begin{bmatrix} m \\ k_1 \\ k_2 \\ \vdots \\ k_{16} \end{bmatrix} = \boxed{c} \quad (mod\ 2)$$

- $DES(K, m_1) \oplus DES(K, m_2) \oplus DES(K, m_3) = ?$

Block Ciphers: DES

- How do we choose S boxes?
- Supose we use S boxes of the following kind:
  - $S_i(x_1, x_2, \ldots, x_6) = (x_2 \oplus x_3, x_1 \oplus x_4 \oplus x_5, x_1 \oplus x_6, x_2 \oplus x_3 \oplus x_6)$
- Do you see a problem using such S boxes?
  - The cipher would be linear.

$$DES(K, m) = \overset{832}{\underset{64}{\boxed{\mathrm{B}}}} \cdot \begin{bmatrix} m \\ k_1 \\ k_2 \\ \vdots \\ k_{16} \end{bmatrix} = \boxed{c} \quad (mod\ 2)$$

- $DES(K, m_1) \oplus DES(K, m_2) \oplus DES(K, m_3) = DES(K, m_1 \oplus m_2 \oplus m_3)$

Block Ciphers: DES

- How do we choose S boxes?
- There are several rules for choosing an S box. Here are a few examples:
  - Should not be chosen randomly.
  - No output bit should be close to a linear function of the input bits.
  - They should be 4-to-1 map.
  - .
  - .

Key Recovery(KR) Attacks on Block Ciphers

KR Attack on Block Ciphers

- <u>Known Plaintext Attack(KPA)</u>: The adversary knows a few pairs $(m_1, c_1), \ldots, (m_q, c_q)$ such that $\forall i, c_i = E(K, m_i)$. The goal is to find K.
- <u>Chosen Plaintext Attack(CPA)</u>: Adversary can pick messages $m_1, \ldots, m_q$ such that it knows their corresponding ciphertexts $c_i = E(K, m_i)$. The goal is to find K.
- The most bruteforce way to find the value of K is to do an Exhaustive Key Search (EKS).
  - $EKS(m, c)$
    - For $K = 0\ to\ 2^{k-1}$
      - If $E(K, m) = c$, then output K
  - Is this guaranteed to give the correct key?

KR Attack on Block Ciphers

- <u>Known Plaintext Attack(KPA)</u>: The adversary knows a few pairs $(m_1, c_1), \ldots, (m_q, c_q)$ such that $\forall i, c_i = E(K, m_i)$. The goal is to find K.
- <u>Chosen Plaintext Attack(CPA)</u>: Adversary can pick messages $m_1, \ldots, m_q$ such that it knows their corresponding ciphertexts $c_i = E(K, m_i)$. The goal is to find K.
- The most bruteforce way to find the value of K is to do an Exhaustive Key Search (EKS).
  - $EKS(m, c)$
    - For $K = 0\ to\ 2^{k-1}$
      - If $E(K, m) = c$, then output K
  - Is this guaranteed to give the correct key?
    - No but usually it does.

KR Attack on Block Ciphers: EKS

- The most bruteforce way to find the value of K is to do an Exhaustive Key Search (EKS).
  - $EKS(m, c)$
    - For $K = 0\ to\ 2^{k-1}$
      - If $E(K, m) = c$, then output K

How long does exhaustive key search take?

DES can be computed at 1.6 Gbits/sec in hardware.

DES plaintext = 64 bits

Chip can perform $(1.6 \times 10^9)/64 = 2.5 \times 10^7$ DES computations per second

Expect EKS to succeed in 2^{55} DES computations, so it takes time

$$\frac{2^{55}}{2.5 \times 10^7} \approx 1.4 \times 10^9 \text{ seconds}$$
$$\approx 45 \text{ years!}$$

Key Complementation $\Rightarrow$ 22.5 years

But this is prohibitive.

Does this mean DES is secure?

KR Attack on Block Ciphers: EKS

- The most bruteforce way to find the value of K is to do an Exhaustive Key Search (EKS).
  - $EKS(m, c)$
    - For $K = 0\ to\ 2^{k-1}$
      - If $E(K, m) = c$, then output K

Differential and linear cryptanalysis

Exhaustive key search is a generic attack: Did not attempt to "look inside" DES and find/exploit weaknesses.

Method	when	q	Type of attack
Differential cryptanalysis	1992	2^{47}	Chosen-message
Linear cryptanalysis	1993	2^{44}	Known-message

But merely storing 2^{44} input-output pairs requires 281 Tera-bytes.

In practice these attacks are prohibitively expensive.

KR Attack on Block Ciphers: EKS

- The most bruteforce way to find the value of K is to do an Exhaustive Key Search (EKS).
  - $EKS(m, c)$
    - For $K = 0\ to\ 2^{k-1}$
      - If $E(K, m) = c$, then output K

- History of attacks on DES:
  - 1992: Biham and Shamir report the first theoretical attack with less complexity than brute force: differential cryptanalysis. However, it requires an unrealistic 2^{47} chosen plaintexts.
  - 1997: The DESCHALL Project breaks a message encrypted with DES for the first time in public. (**Time: 3 months**)
  - 1998: The EFF's DES cracker (Deep Crack) breaks a DES key. (**Time: 56 Hours**)
  - 1999: Together, Deep Crack and distributed.net break a DES key. (**Time:22 hours and 15 minutes**)

KR Attack on Block Ciphers: EKS

- The most bruteforce way to find the value of K is to do an Exhaustive Key Search (EKS).
  - $EKS(m, c)$
    - For $K = 0\ to\ 2^{k-1}$
      - If $E(K, m) = c$, then output K

- History of attacks on DES:
  - 1992: Biham and Shamir report the first theoretical attack with less complexity than brute force: differential cryptanalysis. However, it requires an unrealistic 2^{47} chosen plaintexts.
  - 1997: The DESCHALL Project breaks a message encrypted with DES for the first time in public. (**Time: 3 months**)
  - 1998: The EFF's DES cracker (Deep Crack) breaks a DES key. (**Time: 56 Hours**)
  - 1999: Together, Deep Crack and distributed.net break a DES key. (**Time:22 hours and 15 minutes**)
  - 2006: The FPGA based parallel machine COPACOBANA of the Universities of Bochum and Kiel, Germany, breaks DES in 9 days at $10,000 hardware cost.[19] Within a year software improvements reduced the average time to 6.4 days.
  - 2008: The successor of COPACOBANA, the RIVYERA machine reduced the average time to less than one single day.
- Verdict: The key length is too small even for EKS.
- History: AES becomes effective from 2002.

KR Attack on Block Ciphers: EKS

- 2DES: $\{0,1\}^{112} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ defined by

$$2DES_{K_1K_2}(m) = DES_{K_2}\left(DES_{K_1}(m)\right)$$

- EKS will take 2^{112} DES computations.
- Is there a better way to mount a Key Recovery attack?

K_2	$DES_{K_2}^{-1}(c)$
$00 \ldots 0$	x_0
$00 \ldots 1$	x_1
.	.
$11 \ldots 1$	x_{2^n-1}

Match x from the left table to a y in the right table

K_1	$DES_{K_1}(m)$
$00 \ldots 0$	y_0
$00 \ldots 1$	y_1
.	.
$11 \ldots 1$	y_{2^n-1}

KR Attack on Block Ciphers: EKS

- 2DES: $\{0,1\}^{112} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ defined by

$$2DES_{K_1K_2}(m) = DES_{K_2}\left(DES_{K_1}(m)\right)$$

- EKS will take 2^{112} DES computations.
- Is there a better way to mount a Key Recovery attack?
- This attack takes 2^{57} DES/DES^{-1} computations.
- So the "ëffective" key length for 2DES is 57.

K_2	$DES_{K_2}^{-1}(c)$
00 … 0	x_0
00 … 1	x_1
.	.
11 … 1	x_{2^n-1}

Match x from the left table to a y in the right table

K_1	$DES_{K_1}(m)$
00 … 0	y_0
00 … 1	y_1
.	.
11 … 1	y_{2^n-1}

KR Attack on Block Ciphers: EKS

- 3DES3: $\{0,1\}^{168} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ defined by

$$3DES3_{K_1K_2K_3}(m) = DES_{K_3}\left(DES_{K_2}^{-1}\left(DES_{K_1}(m)\right)\right)$$

- What is "effective" key length with respect to the Meet-in-the-middle attack?

KR Attack on Block Ciphers: EKS

- 3DES3: $\{0,1\}^{168} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ defined by

$$3DES3_{K_1K_2K_3}(m) = DES_{K_3}\left(DES_{K_2}^{-1}\left(DES_{K_1}(m)\right)\right)$$

- What is "effective" key length with respect to the Meet-in-the-middle attack?
  - 113

KR Attack on Block Ciphers: EKS

- DESX: $\{0,1\}^{184} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ defined by

$$DESX_{KK_1K_2}(m) = K_2 \oplus DES_K(K_1 \oplus m)$$

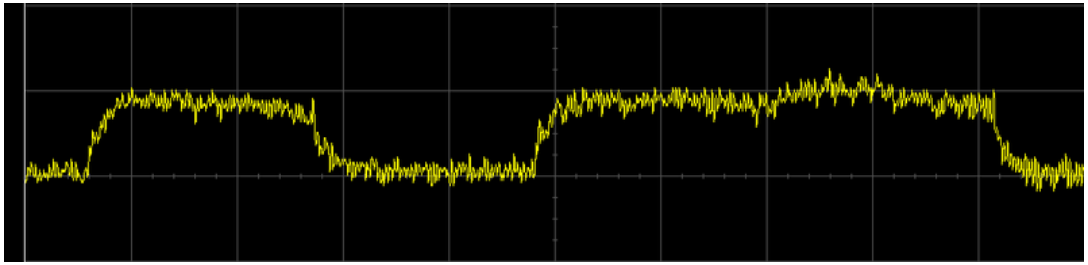
- Key length = 56 + 64 + 64 = 184
- What is "effective" key length with respect to the Meet-in-the-middle attack?
  - 121

Block Ciphers: AES

- AES history:
  - 1998: NIST announces competition for a new block cipher.
    - Requirement:
      - Key length: 128
      - Block length: 128
      - Faster than DES in software.
    - There were15 submissions.
  - 2001: NIST selects Rijndael to be AES.

Side Channel Attacks on Block Ciphers

- Side channel attacks are attacks on the implementation of block ciphers.
- Examples:
  - Analysing time/power/acoustics of encryption/decryption to figure out the secret key.
  - Introducing faults while computation.



- Never design and implement your own block cipher unless you have adequate experience.

End

Acknowledgements:

- Slides 13,14,15,25, and 26 have been borrowed from Mihir Bellare's slides on Cryptography.
- Slides 10,11,12,16,17,18, 19, 20 are taken from lectures slides of Dan Boneh's Cryptography course.