# Lecture Notes

Advanced Discrete Structures

COT 4115.001 S15

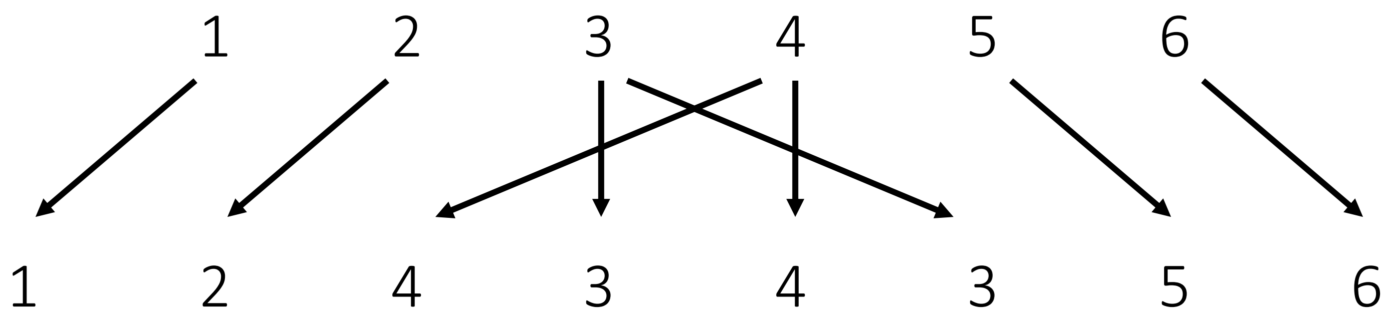
2015-02-10

# Recap

- Simplified DES-like Algorithm
  - <u>Input</u>: Plaintext $L_0R_0$ (12-bit), Key $K$ (9-bit)
  - <u>Rounds</u>: $L_iR_i \leftrightarrow L_{i+1}R_{i+1}$, Round keys $K_i$ (8-bit)
  - <u>Feistel Function</u>: $f: \mathbb{Z}_2^6 \times \mathbb{Z}_2^8 \to \mathbb{Z}_2^6$
    - Expansion $E: \mathbb{Z}_2^6 \to \mathbb{Z}_2^8$
    - Key Mixing $K_i: \mathbb{Z}_2^9 \times \mathbb{Z}_9 \to \mathbb{Z}_2^8$,
      $\bigoplus: \mathbb{Z}_2^8 \times \mathbb{Z}_2^8 \to \mathbb{Z}_2^8$
    - Substitution (S-boxes) $S: \mathbb{Z}_2^4 \times \mathbb{Z}_2^4 \to \mathbb{Z}_2^3 \times \mathbb{Z}_2^3$

# Recap

## Expansion function $\quad E: \mathbb{Z}_2^6 \rightarrow \mathbb{Z}_2^8$




## S-boxes

$S_1$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 101 | 010 | 001 | 110 | 011 | 100 | 111 | 000 |
| 001 | 100 | 110 | 010 | 000 | 111 | 101 | 011 |

$S_2$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 100 | 000 | 110 | 101 | 111 | 001 | 011 | 010 |
| 101 | 011 | 000 | 111 | 110 | 010 | 001 | 100 |

# Recap

- <u>Encryption</u>:
  - $L_0 R_0 \to L_1 R_1 \to \cdots \to L_n R_n$ with keys $K_1, K_2, \dots, K_n$ using:
    - $[L_i]\ [R_i] \ \to \ [R_i]\ [L_i \oplus f(R_i, K_{i+1})] = [L_{i+1}]\ [R_{i+1}]$
- <u>Decryption</u>:
  - Swap the blocks $L_n$ and $R_n$, and use the encryption algorithm with keys $K_n, K_{n-1}, \dots, K_1$
    - $[R_{i+1}]\ [L_{i+1}] \ \to \ [L_{i+1}]\ [R_{i+1} \oplus f(L_{i+1}, K_{i+1})] = [R_i]\ [L_i]$
  - When you get to $R_0 L_0$, swap the blocks back to get $L_0 R_0$

# DES-type Algorithm - Main

<u>Message</u>: 110011010101 <u>Key</u>: 010101110

| $i$ | $\boldsymbol{K_i}$ | $[\boldsymbol{L_i}]\ [\boldsymbol{R_i}]$ | $[\boldsymbol{R_i}]\ [\boldsymbol{L_i}]$ |
|---|---|---|---|
| 0 | -- | [110011] [010101] | |
| 1 | 01010111 | | |

# DES-type Algorithm - Encryption

Key: $0 1 0 1 0 1 1 1 0$ Message: $1 1 0 0 1 1 0 1 0 1 0 1$

Encryption ($i = 0$):

$$[L_0] = 110011 \qquad [R_0] = 010101$$

Encryption ($i = 1$):

$$[L_1] \ [R_1] \ = \ [R_0] \ [L_0 \oplus f(R_0, K_1)]$$

$K_1 = 01010111$ Just need $f(R_0, K_1)$

# DES-type Algorithm - Encryption

$$[R_0] = 010101 \qquad \text{and} \qquad K_1 = 01010111$$

Compute $\boldsymbol{f(R_0, K_1)}$:

1. <u>Expansion</u>: $E(R_0) = E(01\underline{01}01) = 01\underline{1010}01$

   $(12\underline{34}56 \rightarrow 12\underline{4343}56)$

2. <u>Key Mixing</u>: $E(R_0) \oplus K_1$

$$= 01101001 \oplus 01010111$$

$$= 00111110$$

# DES-type Algorithm - Encryption

$$E(R_0) \oplus K_1 = 00111110 \implies 0011 \quad 1110$$

3. <u>Substitution</u>:

$\mathbf{0011}$: First block $\rightarrow$ $S_1$,
First bit $\rightarrow$ first row,
Next 3 bits $\rightarrow$ column $011$

$\mathbf{1110}$: Second block $\rightarrow S_2$
First bit $\rightarrow$ second row,
Next 3 bits $\rightarrow$ column $110$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $S_1$ | 101 | 010 | 001 | 110 | 011 | 100 | 111 | 000 |
| | 001 | 100 | 110 | 010 | 000 | 111 | 101 | 011 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $S_2$ | 100 | 000 | 110 | 101 | 111 | 001 | 011 | 010 |
| | 101 | 011 | 000 | 111 | 110 | 010 | 001 | 100 |

$$0011 \;\; 1110 \implies 110 \;\; 001 \qquad\qquad f(R_0, K_1) = 110001$$

# DES-type Algorithm - Encryption

We Know:

$[L_0] = 110011, \qquad [R_0] = 010101, \qquad f(R_0, K_1) = 110001$

Round $i = 1$:

$$[L_0]\ [R_0] \ \rightarrow \ [R_0]\ [L_0 \oplus f(R_0, K_1)] = [L_1]\ [R_1]$$

$[L_1] = 010101$

$[R_1] = L_0 \oplus f(R_0, K_1) = 110011 \oplus 110001 = 000010$

# DES-type Algorithm - Main

<u>Message</u>: 110011010101 <u>Key</u>: 010101110

| $i$ | $K_i$ | $[L_i]$ $[R_i]$ | $[R_i]$ $[L_i]$ |
|---|---|---|---|
| 0 | -- | [110011] [010101] | |
| 1 | 01010111 | [010101] [000010] | |
| 2 | 10101110 | | |

# DES-type Algorithm - Encryption

$\boxed{i = 2}$ $[L_1] = 010101$ $[R_1] = 000010$ $K_2 = 10101110$

<u>Rule</u>: $[L_1]\ [R_1] \ \rightarrow\ [R_1]\ [L_1 \oplus f(R_1, K_2)]$

<u>Expansion</u>: $E(R_1) = 00\underline{0000}10$

<u>Key Mixing</u>: $E(R_1) \oplus K_2 = 00000010 \oplus 10101110 = 10101100$

<u>Substitution</u>: $S_1: 1010 \rightarrow 110$ $S_2: 1100 \rightarrow 110,$ $f(R_1, K_2) = 110110$

$[L_2] = 000010$

$[R_2] = L_1 \oplus f(R_1, K_2) = 010101 \oplus 110110 = 100011$

# DES-type Algorithm - Main

<u>Message</u>: 110011010101 <u>Key</u>: 010101110

| $i$ | $K_i$ | $[L_i]$ $[R_i]$ | $[R_i]$ $[L_i]$ |
|---|---|---|---|
| 0 | -- | [110011] [010101] | |
| 1 | 01010111 | [010101] [000010] | |
| 2 | 10101110 | [000010] [100011] | |
| 3 | 01011100 | | |

# DES-type Algorithm - Encryption

$\boxed{i = 3}$ $[L_2] = 000010$ $[R_2] = 100011$ $K_3 = 01011100$

Rule: $[L_2]\ [R_2] \ \rightarrow \ [R_2]\ [L_2 \oplus f(R_2, K_3)]$

Expansion: $E(R_2) = 10\underline{0000}11$

Key Mixing: $E(R_2) \oplus K_3 = 10000011 \oplus 01011100 = 11011111$

Substitution: $S_1: 1101 \rightarrow 111 \quad S_2: 1111 \rightarrow 100, \ \ f(R_1, K_2) = 111100$

$[L_3] = 000010$

$[R_3] = L_2 \oplus f(R_2, K_3) = 000010 \oplus 111100 = 111110$

# DES-type Algorithm - Main

<u>Message</u>: 110011010101      <u>Key</u>: 010101110

| $i$ | $\boldsymbol{K_i}$ | $[\boldsymbol{L_i}]$ $[\boldsymbol{R_i}]$ | $[\boldsymbol{R_i}]$ $[\boldsymbol{L_i}]$ |
|---|---|---|---|
| 0 | -- | [110011] [010101] | |
| 1 | 01010111 | [010101] [000010] | |
| 2 | 10101110 | [000010] [100011] | |
| 3 | 01011100 | [100011] [111110] | |

# DES-type Algorithm - Main

**Message**: 110011010101  **Key**: 010101110

| $i$ | $\boldsymbol{K_i}$ | $[\boldsymbol{L_i}]\ [\boldsymbol{R_i}]$ | | $[\boldsymbol{R_i}]\ [\boldsymbol{L_i}]$ |
|---|---|---|---|---|
| 0 | -- | [110011] [010101] | | |
| 1 | 01010111 | [010101] [000010] | | |
| 2 | 10101110 | [000010] [100011] | | |
| 3 | 01011100 | [100011] [111110] | $\Rightarrow$ | [111110] [100011] |

# DES-type Algorithm - Decryption

$\boxed{i = 3}$ $[L_3] = 100011$ $[R_3] = 111110$ $K_3 = 01011100$

Rule: $[R_3]\ [L_3] \ \rightarrow\ [L_3]\ [R_3 \oplus f(L_3, K_3)]$

Expansion: $E(L_3) = 10\underline{0000}11$

Key Mixing: $E(L_3) \oplus K_3 = 10000011 \oplus 01011100 = 11011111$

Substitution: $S_1: 1101 \rightarrow 111 \quad S_2: 1111 \rightarrow 100, \ \ f(R_1, K_2) = 111100$

$[R_2] = 100011$

$[L_2] = R_3 \oplus f(L_3, K_3) = 111110 \oplus 111100 = 000010$

# DES-type Algorithm - Main

**Message**: 110011010101      **Key**: 010101110

| $i$ | $\boldsymbol{K_i}$ | $[\boldsymbol{L_i}]\ [\boldsymbol{R_i}]$ | $[\boldsymbol{R_i}]\ [\boldsymbol{L_i}]$ |
|---|---|---|---|
| 0 | -- | [110011] [010101] | |
| 1 | 01010111 | [010101] [000010] | |
| 2 | 10101110 | [000010] [100011] | [100011] [000010] |
| 3 | 01011100 | [100011] [111110] | [111110] [100011] |

# DES-type Algorithm - Main

**Message**: 110011010101 **Key**: 010101110

| $i$ | $K_i$ | $[L_i]$ $[R_i]$ | $[R_i]$ $[L_i]$ |
|---|---|---|---|
| 0 | -- | [110011] [010101] | [010101] [110011] |
| 1 | 01010111 | [010101] [000010] | [000010] [010101] |
| 2 | 10101110 | [000010] [100011] | [100011] [000010] |
| 3 | 01011100 | [100011] [111110] | [111110] [100011] |

# DES-type Algorithm - Main

**Message**: 110011010101 **Key**: 010101110

| $i$ | $K_i$ | $[L_i]$ $[R_i]$ | | $[R_i]$ $[L_i]$ |
|---|---|---|---|---|
| 0 | -- | [110011] [010101] | ⟵ | [010101] [110011] |
| 1 | 01010111 | [010101] [000010] | | [000010] [010101] |
| 2 | 10101110 | [000010] [100011] | | [100011] [000010] |
| 3 | 01011100 | [100011] [111110] | | [111110] [100011] |

Basic Number Theory - Section 3.5

# MODULAR EXPONENTIATION

# Modular Exponentiation

Computing

$$x^a \pmod{n}$$

<u>Options</u>:

1. Compute $x^a$ then consider $\bmod n$
   - $x^a$ may be huge (too big to store)

2. Represent $x^a = x^{b_1+b_2+\cdots+b_m} = x^{b_1}x^{b_2}\ldots x^{b_m}$ and evaluate each $x^{b_i} \pmod{n}$ individually. Multiply the results together to recover $x^a \pmod{n}$
   - If $m$ is big, too many numbers $b_1, b_2, \ldots, b_m$

# Modular Exponentiation

- Note that
$$x^{b^e} = x^{b \cdot b^{e-1}} = \left(x^{b^{e-1}}\right)^b,$$
so if $x^{b^{e-1}}$ is known, then $x^{b^e}$ is easy to compute

- For $x^a \pmod{n}$, represent $a$ in some smaller base system, i.e.,
$$a = c_n b^n + c_{n-1} b^{n-1} + \cdots + c_1 b + c_0,$$
and compute each $x^{b^e} \pmod{n}$ from $x^{b^{e-1}} \pmod{n}$

- Typical to use binary, i.e., $b = 2$ and each $c_0 \in \mathbb{Z}_2$.

# Modular Exponentiation

Example:

$$123^{45} \pmod{67}$$

1. Represent $45$ in binary, $45_{10} = 101101_2$, so
$$\begin{aligned} 123^{45} &= 123^{2^5+2^3+2^2+2^0} \\ &= \left(123^{2^5}\right)\left(123^{2^3}\right)\left(123^{2^2}\right)\left(123^{2^0}\right) \end{aligned}$$

2. Compute $123^{2^e} \pmod{67}$ from $123^{2^{e-1}} \pmod{67}$

# Modular Exponentiation

Example:

$$123^{45} \pmod{67}$$

$$123^{2^0} = 123^1 \equiv 56 \pmod{67}$$
$$123^{2^1} = \left(123^{2^0}\right)^2 \equiv 56^2 = 3{,}136 \equiv 54 \pmod{67}$$
$$123^{2^2} = \left(123^{2^1}\right)^2 \equiv 54^2 = 2{,}916 \equiv 35 \pmod{67}$$
$$123^{2^3} = \left(123^{2^2}\right)^2 \equiv 35^2 = 1{,}225 \equiv 19 \pmod{67}$$
$$123^{2^4} = \left(123^{2^3}\right)^2 \equiv 19^2 = 361 \equiv 26 \pmod{67}$$
$$123^{2^5} = \left(123^{2^4}\right)^2 \equiv 26^2 = 676 \equiv 6 \pmod{67}$$

# Modular Exponentiation

Example:

$$123^{45} \equiv 123^{2^5} 123^{2^3} 123^{2^2} 123^{2^0} \pmod{67}$$
$$\equiv \quad (6) \quad (19) \quad (35) \quad (56) \pmod{67}$$
$$\equiv 62 \pmod{67}$$

$$123^{2^0} \equiv 56 \pmod{67}$$
$$123^{2^1} \equiv 54 \pmod{67}$$
$$123^{2^2} \equiv 35 \pmod{67}$$
$$123^{2^3} \equiv 19 \pmod{67}$$
$$123^{2^4} \equiv 26 \pmod{67}$$
$$123^{2^5} \equiv 6 \pmod{67}$$

# Modular Exponentiation

Even though

$$123^{45} = 11{,}110{,}408{,}185{,}131{,}956{,}285{,}910{,}790{,}587{,}176{,}451{,}918{,}559{,}153{,}212{,}268{,}021{,}823{,}629{,}073{,}199{,}866{,}111{,}001{,}242{,}743{,}283{,}966{,}127{,}048{,}043$$

we never computed a number bigger than

$$66^2 = 4{,}356$$

and only used $8$ multiplications.