Block Ciphers

Sugata Gangopadhyay

Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Symmetric cryptosystems

- The cryptosystems in which the decryption key can be derived efficiently from the knowledge of the encryption key are called symmetric cryptosystems.

- Symmetric cryptosystems are divided into stream ciphers are block ciphers.

- Presently, we will discuss block ciphers.

Block ciphers

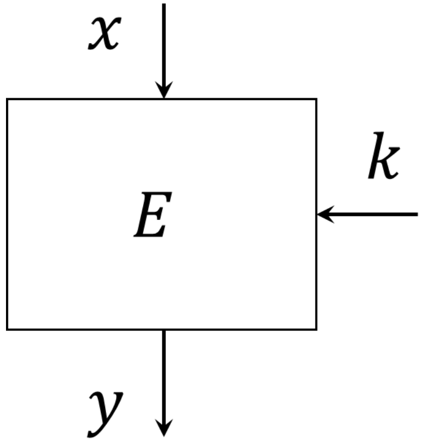
- A block cipher is a cryptosystem that transforms an n-bit plaintext block to any n-bit ciphertext block using an m-bit key.

- A block cipher is represented by a function

$$E\colon \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$$

where $\mathcal{P} = \mathcal{C} = \{0,1\}^n$ and $\mathcal{K} = \{0,1\}^m$.

- The adjacent figure is a diagram of a block cipher.



Iterated Block Ciphers: the Key Schedule

- Let K be a random binary key of some specified length.

- K is used to construct NR round keys (also called subkeys), which are denoted by $K^1, \dots, K^{NR}$.

- The list of round keys $K^1, \dots, K^{NR}$ is called the key schedule.

- The key schedule is constructed from K by using a fixed public algorithm.

Iterated Block Ciphers: the round function

- The round function, say g, takes two inputs: a round key (K^r) and a current state denoted by w^{r-1}.

- The initial state w^0 is defined to be the plaintext, x.

- The ciphertext y is defined to be the state after all NR rounds.

The round function: encryption

- The encryption operation is carried out as follows:

$$
\begin{array}{ll}
w^0 \leftarrow x & \\
w^1 \leftarrow g(w^0, \kappa^1) & \\
w^2 \leftarrow g(w^1, \kappa^2) & \\
\vdots \qquad\qquad \vdots & \\
w^{NR-1} \leftarrow g(w^{NR-2}, K^{NR-1}) & \\
w^{NR} \leftarrow g(w^{NR-1}, K^{NR}) & \\
y \leftarrow w^{NR} &
\end{array}
$$

The round function: decryption

- For decryption to be possible the function g must have the property that it is injective (one-to-one) if its second argument is fixed.

$$
\begin{array}{l}
w^{NR} \leftarrow y \\
w^{NR-1} \leftarrow g^{-1}(w^{NR}, \kappa^{NR}) \\
\vdots \qquad\qquad\qquad\quad \vdots \\
w^{1} \leftarrow g^{-1}(w^{2}, \kappa^{2}) \\
w^{0} \leftarrow g^{-1}(w^{1}, \kappa^{1}) \\
x \leftarrow w^{0}
\end{array}
$$

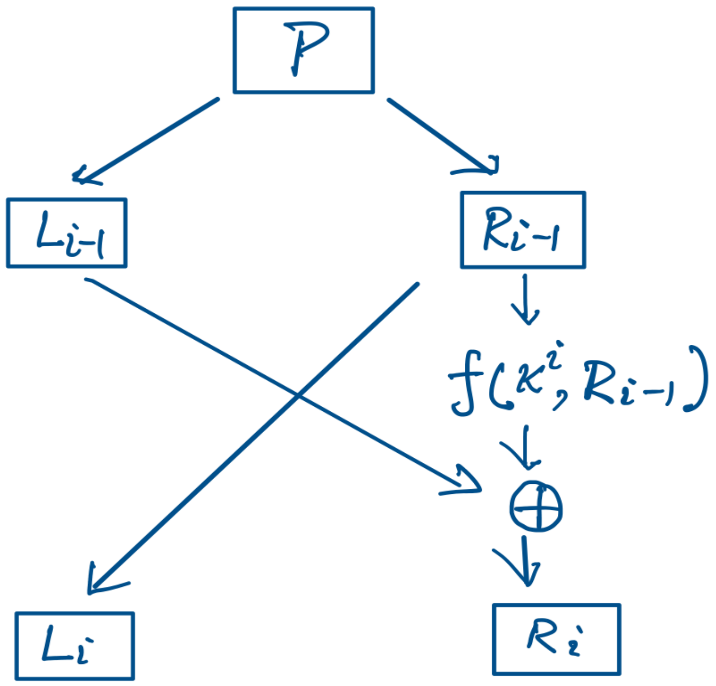
Feistel Cryptosystem

- A Feistel cryptosystem (also known as a Feistel network of system) is a block cryptosystem determined by the following components:
  - The block size $2t$ (an even number) and a key size N
  - The number of rounds NR (a positive integer)
  - A key schedule: A mechanism for generating NR round keys $\kappa^1, \kappa^2, \dots, \kappa^{NR}$ from the single cryptosystem key κ.
  - A round key function, f_{κ^i}, for each round key κ^i, which inputs any t-bit string R, and outputs another t-bit string $f_{\kappa^i}(R)$.

Encryption and Decryption

- Encryption
  - Given a plaintext P, a bit string of length $2t$, we first split P into two bit strings of length t: $P = (L_0, R_0)$, with L_0 being the left half, and R_0 being the right half.
  - We then proceed through NR rounds for the following transformations:
    For round $i = 1$ to NR: $L_i = R_{i-1},\ R_i = L_{i-1} \oplus f_{\kappa^i}(R_{i-1})$
  - The ciphertext will then be $C = (R_{NR}, L_{NR})$
- Decryption
  - We feed the ciphertext C back into the above encryption process, the only change being that the keys are used in the reverse order.

One round of Feistel network



A simple three round Feistel cryptosystem

- The block size = 8. The key length = 12. The number of rounds = 3.
- The key scheduling algorithm
  - For each length-12 bit string $\kappa = k_1 k_2 \ldots k_{12}$ the round keys are

$$\begin{aligned} \kappa^1 &= k_1 \cdots \mathrm{k}_4 \oplus k_5 \cdots k_8 \\ \kappa^2 &= k_5 \cdots \mathrm{k}_8 \oplus k_9 \cdots k_{12} \\ \kappa^3 &= k_9 \cdots \mathrm{k}_{12} \oplus k_1 \cdots k_4 \end{aligned}$$

- The round key function $f_{\kappa^i}(R)$ is simply obtained by XORing an a 4-bit input string R with the round key κ^i.

Exercise

- With a 12-bit system key $\kappa = \mathrm{ABC}$ (represented in hex form), use this Feistel cryptosystem to encrypt the 8-bit plaintext $P = \mathrm{DF}$ (represented in hex form)
- Perform the corresponding decryption to the ciphertext that resulted that resulted from P.

Data Encryption Standard (DES)

- In 1973 the US government initiated a competition to create an efficient cryptographic protocol that would be suitably secure for financial transactions and business communications to serve as a national standard.
- The public announcement was made by the National Bureau of Standards (NBS) which is now known as the National Institute of Standards and Technology (NIST).
- The outcome of this competition was the selection of the cipher Lucifer designed by IBM scientist lead by Horst Feistel (1915-1990).
- This cipher is called DES or Data Encryption Standard.

Data Encryption Standard (DES)

- The detailed description of DES is out of scope for these lectures.
- DES is a Feistel cryptosystem.
- The general structure of Feistel Cryptosystems is described in the previous slides.

Substitution-Permutation Network (SPN)

- A plaintext and ciphertext both are binary vectors of length ℓm, i.e., they are elements of $\{0,1\}^{\ell m}$. The integer ℓm is said to be the block-length of the cipher. Two components of an SPN are
  - A substitution function that is also known as an S-box:
    $$\pi_S\colon \{0,1\}^{\ell} \rightarrow \{0,1\}^{\ell}$$
  - A permutation function: $\pi_P\colon [\,\ell m] \rightarrow [\ell m]$ where $[\ell m] = \{1, \ldots, \ell m\}$.

Substitution-Permutation Network

- Let ℓ, m, and NR be positive integers, let π_S: $\{0,1\}^{\ell} \to \{0,1\}^{\ell}$ be a permutation, and π_P: $[\ell m] \to [\ell m]$ be a permutation. Let
$$\mathcal{P} = \mathcal{C} = \{0,1\}^{\ell m}.$$

- Let $\mathcal{K} \subseteq \left(\{0,1\}^{\ell m}\right)^{NR+1}$ consists of all possible key schedules that could be derived from an initial key K using the key scheduling algorithm. For a key schedule $(K^1, \ldots, K^{NR+1})$, we encrypt the plaintext x using the algorithm in the next slide.

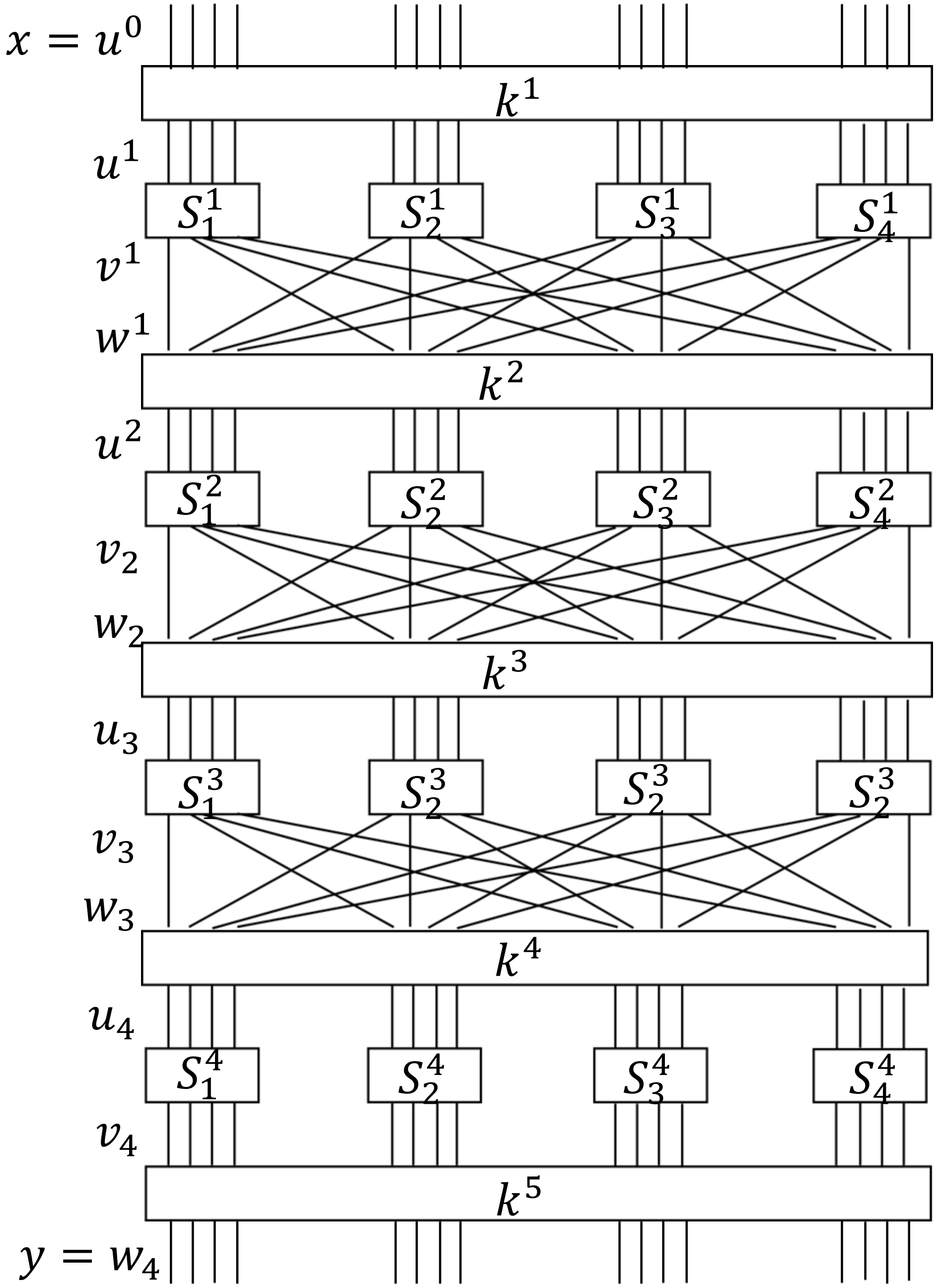
Substitution-Permutation Network

- The plaintext $x = x_{\langle 1 \rangle} || \, x_{\langle 2 \rangle} || \cdots || x_{\langle m \rangle}$ where $x_{\langle i \rangle} \in \{0,1\}^{\ell}$.
- The rth round input is $u^r = u^r_{\langle 1 \rangle} || \, u^r_{\langle 2 \rangle} || \cdots || u^r_{\langle m \rangle}$ where $u^r_{\langle i \rangle} \in \{0,1\}^{\ell}$.
- Substitution output $v^r = v^r_{\langle 1 \rangle} || \, v^r_{\langle 2 \rangle} || \cdots || v^r_{\langle m \rangle}$ where $v^r_{\langle i \rangle} \in \{0,1\}^{\ell}$.
- Permutation output $w^r = w^r_{\langle 1 \rangle} || \, w^r_{\langle 2 \rangle} || \cdots || w^r_{\langle m \rangle}$ where $w^r_{\langle i \rangle} \in \{0,1\}^{\ell}$.
- The ciphertext $y = y_{\langle 1 \rangle} || \, y_{\langle 2 \rangle} || \cdots || y_{\langle m \rangle}$ where $y_{\langle i \rangle} \in \{0,1\}^{\ell}$.

Algorithm: Substitution-Permutation Network

Algorithm: Substitution-Permutation Network

```
Input  : $x, \pi_S, \pi_P, (K^1, \dots, K^{NR+1})$
Output : $y$
$w^0 \leftarrow x$
for $r = 1$ to $NR - 1$
    $u^r \leftarrow w^{r-1} \bigoplus K^r$
    for $i = 1$ to $m$
        $v^r_{\langle i \rangle} \leftarrow \pi_S\left(u^r_{\langle i \rangle}\right)$
    $w^r \leftarrow \left(v^r_{\langle \pi_P(1) \rangle}, \dots, v^r_{\pi_P(\ell m)}\right)$
$u^{NR} \leftarrow w^{NR-1} \bigoplus K^{NR}$
for $i = 1$ to $m$
    $v^{NR}_{\langle i \rangle} \leftarrow \pi_S\left(u^{NR}_{\langle i \rangle}\right)$
$y \leftarrow v^{NR} \bigoplus K^{NR+1}$
```


$x = u^0$
k^1
u^1
S_1^1
S_2^1
S_3^1
S_4^1
v^1
w^1
k^2
u^2
S_1^2
S_2^2
S_3^2
S_4^2
v_2
w_2
k^3
u_3
S_1^3
S_2^3
S_2^3
S_2^3
v_3
w_3
k^4
u_4
S_1^4
S_2^4
S_3^4
S_4^4
v_4
k^5
$y = w_4$

Modes of operations for block ciphers

- Electronic Codebook (ECB)

- Cipher-block Chaining (CBC)

- Cipher Feedback (CFB)

- Output Feedback (OFB)

Modes of operations for block ciphers

Notation:

- We denote a sequence of plaintext blocks by $P_1, P_2, P_3, \cdots,$; each will be a block of ℓ bits (ℓ = the block length).
- The encryption mapping of the particular block cipher being employed is denoted as E_κ, and its inverse (the decryption mapping) is denoted as $D_\kappa \triangleq E_\kappa^{-1}$.
- The corresponding blocks of ciphertexts that the modes of operation will send are denoted as $C_1, C_2, C_3, \cdots$, each of length ℓ.

Electronic Codebook (ECB)

- Encrypt consecutive blocks (with the same key) and transmit the corresponding ciphertext blocks.
- This is called the electronic codebook (ECB) mode. This is represented by the equation
$$C_i = E_\kappa(P_i), \qquad i = 1, 2, 3, \cdots$$
where P_i is the plaintext, C_i is the ciphertext, and κ is the key.
- To decrypt, the recipient need only apply the inverse encryption mapping to the ciphertext blocks: $P_i = D_\kappa(C_i)$.

Electronic Codebook (ECB) Mode

- ECB mode is susceptible to a codebook attack.
- In a codebook attack:
  - The adversary collects and analyzes large sets of intercepted cipher blocks with whatever tools and additional information might be available, such as frequency analysis, cribs (known plaintext/ciphertext pairs).
  - Once certain cipher-block is ascertained, they are entered into a codebook that can be used to check subsequent messages for partial decryptions without actually having the key.
- To avoid such attacks, it is best to use a mode that adds some additional noise in the processed blocks so that identical plaintext blocks need not be processed into identical cipher blocks.

Cipher-block Chaining (CBC) Mode

- The cipher-block chaining (CBC) mode first XORs each plaintext block P_i with the previously produced ciphertext block C_{i-1}.
- The equation for CBD mode is

$$C_i = E_\kappa(P_i \oplus C_{i-1}), \qquad i = 1, 2, 3, \cdots$$

- The zeroth ciphertext block C_0 is missing in this scheme so far.
- If we use the same C_0 for repeated transmission, then the corresponding first ciphertext blocks C_1 would be susceptible to a codebook attack.
- The zeroth ciphertext block is randomly produced and sent to the receiver without encryption at the time of the start of each new transmission.
- Decryption

$$P_i = D_\kappa(C_i) \oplus C_{i-1}, \qquad i = 1, 2, 3, \cdots$$

Example

- Consider the following block encryption function $E_\kappa = E$ on 2-bit blocks (so the block size if $\ell = 2$) that is defined as follows:

	Block Encryption function			
P	00	01	10	11
$E(P)$	10	00	11	01

For the plaintext: 1010100011

a) Determine the corresponding ciphertext sequence that gets transmitted if electronic codebook (ECB) mode is used.

b) Determine the corresponding ciphertext sequence that gets transmitted if cipher-block chaining mode is used with seed $C_0 = 10$.

Cipher Feedback (CFB) mode

- ECB and CBC modes process entire ℓ-bit blocks at a time, the CFB mode works on smaller subblocks of any size k, where $k \mid \ell$.
- A typical value of $k = 8$, so that each subblock represent one of the 256 ASCII characters.
- Since the CFB mode operates on subblocks of size k, each plaintext block P_i is decomposed into its $n = \frac{\ell}{k}$ subblocks:

$$P_i = p_i^1 p_i^2 \cdots p_i^n, \qquad\qquad i = 1, 2, 3, \cdots$$

- Thus each p_j^m represents a single k-bit subblock.

Cipher Feedback (CFB) Mode Encryption

Assume that we have a string of a certain number of plaintext subblocks $p_1, p_2, p_3, \cdots$, that we need to encrypt and send. The resulting stream of cipher subblocks that the algorithm produces will be denoted as $c_1, c_2, c_3, \cdots$. The encryption mapping E_κ works with strings of n k-bit subblocks.

Cipher Feedback (CFB) Mode Encryption

Cipher Feedback (CFB) Mode Encryption

Step 1. Generate an ℓ-bit shift register S_1. Random generation is recommended. Initialize subblock counter $i = 1$.

Step 2. Encrypt $S_i \to E_\kappa(S_i)$, let T_i be the leftmost k-bit subblock of $E_\kappa(S_i)$, and let R_i be the string of the rightmost $n - 1$ subblocks of the shift register S_i.

Step 3. Set $c_i = p_i \oplus T_i$, update the next shift register as $S_{i+1} = R_i || c_i$, and update $i \to i + 1$.

Step 4. Return to Step 2 until all plaintext subblocks have been encrypted.

Example: Suppose that we are given a plaintext `101110`. Determine the ciphertext that gets transmitted if the encryption function is as given below.

Block Encryption function				
P	00	01	10	11
$E(P)$	10	00	11	01

Cipher Feedback (CFB) Mode Decryption

Cipher Feedback (CFB) Mode Decryption

Step 1. Initialize subblock counter $i = 1$.

Step 2. Encrypt $S_i \to E_\kappa(S_i)$, let T_i be the leftmost k-bit subblock of $E_\kappa(S_i)$, and let R_i be the string of the rightmost $n - 1$ subblocks of the shift register S_i.

Step 3. Define $p_i = c_i \oplus T_i$, update the next shift register as $S_{i+1} = R_i || c_i$, and update $i \to i + 1$.

Step 4. Return to Step 2 until all ciphertext subblocks have been decrypted.

Output Feedback (OFB) Mode

- The CFB mode is a stream cipher protocol that can be used with any block cipher.
- If any error were to be introduced in transmitting any ciphertext subblock, the error would propagate through the stream to corrupt the next $n-1$ decrypted characters, after which the corrupted stream would be flushed from the system.
- Output Feedback (OFB) mode is a more robust algorithm.

Out Feedback (OFB) Mode Encryption

Output Feedback (OFB) Mode Encryption

Step 1. Generate an ℓ-bit shift register S_1. Random generation is recommended. Initialize subblock counter $i = 1$.

Step 2. Encrypt $S_i \to E_\kappa(S_i)$, let T_i be the leftmost k-bit subblock of $E_\kappa(S_i)$, and let R_i be the string of the rightmost $n - 1$ subblocks of the shift register S_i.

Step 3. Define $c_i = p_i \oplus T_i$, update the next shift register as $S_{i+1} = R_i || T_i$, and update $i \to i + 1$.

Step 4. Return to Step 2 until all plaintext subblocks have been encrypted.

Cryptanalytic Attack Model

The following are the cryptanalytic attack model:

- Ciphertext only attack
- Known plaintext attack
- Chosen plaintext attack
- Chosen ciphertext attack

Ciphertext only attack

- The adversary possesses a string of ciphertext.

- Example: the attacks on classical ciphers.

Known plaintext attack

- The adversary possesses a string of plaintext, $\mathbf{x}$, and the corresponding ciphertext $\mathbf{y}$.

- In the process, several plaintext-ciphertext pairs are known to the adversary.

- The goal of the adversary is to find the unknown keys used by the legitimate intended participants of the communication network.

Chosen plaintext attack

- The adversary has the capacity of choosing any plaintext and get it encrypted by the sender.

- In other word, the adversary has at least temporary access to the encrypting device. Hence, he can choose a plaintext string, $\mathbf{x}$, and construct the corresponding ciphertext string $\mathbf{y}$.

Chosen Ciphertext Attack

- The adversary has the capacity of choosing any ciphertext and get it decrypted by the receiver.

- In other word, the adversary has at least temporary access to the decrypting device. However, the key remains unknown to the adversary.

Examples

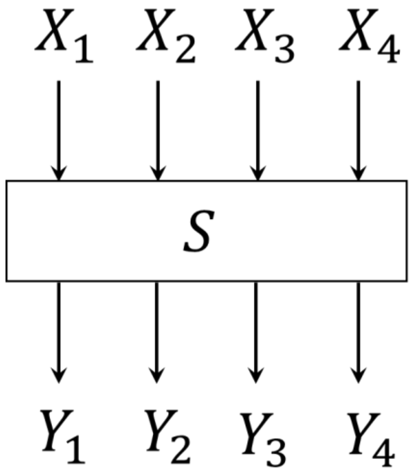
- Linear approximation attack was proposed by Matsui in 1993 to cryptanalyze DES. This is a known plaintext attack.

- Differential attack was proposed by Dinur and Shamir in 1991. This is a chosen ciphertext attack developed to attack DES.

- Over the last three decades, these attacks have evolved to attack stream ciphers.

Probabilistic Formulation

- In the probabilistic model of a cryptosystem:
  - The input variables and output variables are considered as random variables defined on $\{0, 1\}$.
  - $\Pr[X_i = 0] = p_i$ where $0 \leq p_i \leq 1$ is said to be the probability distribution of X_i.
  - $\epsilon_i = p_i - \frac{1}{2}$ is said to the the bias of X_i; $-\frac{1}{2} \leq \epsilon_i \leq \frac{1}{2}$.



- The adjacent figure demonstrates this model for a single S-box.

Independence of random variables

- In general, let $X = (X_1, \dots, X_n), Y = (Y_1, \dots, Y_m)$ be random variables corresponding to the input and output of an $n \times m$ S-box.
- It is reasonable to assume that the random variables $X_1, \dots, X_n$, are independent random variables.

- X_i, X_j are independent then
  - $\Pr[X_i = 0, X_j = 0] = p_i\, p_j$
  - $\Pr[X_i = 0, X_j = 1] = p_i(1 - p_j)$
  - $\Pr[X_i = 1, X_j = 0] = (1 - p_i)p_j$
  - $\Pr[X_i = 1, X_j = 1] = (1 - p_i)(1 - p_j)$.
- The distribution of the discrete random variable $X_i \oplus X_j$ is:
  - $\Pr[X_i \oplus X_j = 0] = p_i p_j + (1 - p_i)(1 - p_j)$
  - $\Pr[X_i \oplus X_j = 1] = p_i(1 - p_j) + (1 - p_i)p_j$.

Piling-up Lemma

Let $\epsilon_{\{i_1,\dots\, i_k\}}$ denote the bias of the random variable $X_{i_1} \oplus \cdots \oplus\ X_{i_k}$ Then $\epsilon\ _{i_1,\dots,i_k} =\ 2^{k-1}\prod_{j=1}^{k}\epsilon_{i_j}$ where ϵ_{i_j} is the bias of the random variable X_{i_j}, for $j\ =\ 1,\dots,k$.

- $\Pr\left[X_{i_1} \oplus\ X_{i_2} =\ 0\right] =\ p_{i_1}\ p_{i_2} +\ (1-p_{i_1})(1-p_{i_2})$
  $$= \left(\frac{1}{2}+\epsilon_{i_1}\right)\left(\frac{1}{2}+\epsilon_{i_2}\right)+\left(\frac{1}{2}-\epsilon_{i_1}\right)\left(\frac{1}{2}-\epsilon_{i_2}\right) = \frac{1}{2}+2\epsilon_{i_1}\epsilon_{i_2}$$
- $\Pr\left[X_{i_1} \oplus X_{i_1} \oplus X_{i_3} = 0\right] = \Pr\left[\left(X_{i_1} \oplus X_{i_2}\right) \oplus X_{i_3}\right]$
  $$= \left(\frac{1}{2}+2\epsilon_{i_1}\epsilon_{i_2}\right)\left(\frac{1}{2}+\epsilon_{i_3}\right)+\left(\frac{1}{2}-2\epsilon_{i_1}\epsilon_{i_2}\right)\left(\frac{1}{2}-\epsilon_{i_3}\right) = \frac{1}{2}+2^2\epsilon_{i_1}\epsilon_{i_2}\epsilon_{i_3}$$

S-box Analysis: Linear Approximation

- Suppose that $S\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ is an $n \times m$ S-box, or equivalently an (n, m)-function.

- Let $X = (X_1, \dots, X_n)$ and $Y = (Y_1, \dots, Y_m)$ denote sequences of random variables that correspond to input and output, respectively.

- For each pair $(a, b) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ construct the following sum

$$a \cdot X \oplus b \cdot Y = (\bigoplus_{i=1}^{n} a_i X_i) \oplus (\bigoplus_{i=1}^{m} b_i Y_i).$$

S-box Analysis: Linear Approximation table

- Suppose $x = (x_1, \dots x_n)$ and $y = (y_1, \dots, y_m)$.
- $N_L(a, b) = \#\{ (x, y) : y = \pi_S(x), a \cdot x \oplus b \cdot y = 0 \}$
- For a 4×4 S-box

$$N_L(a, b) = \#\left\{ (x, y) \colon y = \pi_{S(x)}, (\bigoplus_{i=1}^{4} a_i x_i) \oplus (\bigoplus_{i=1}^{4} b_i y_i) = 0 \right\}$$

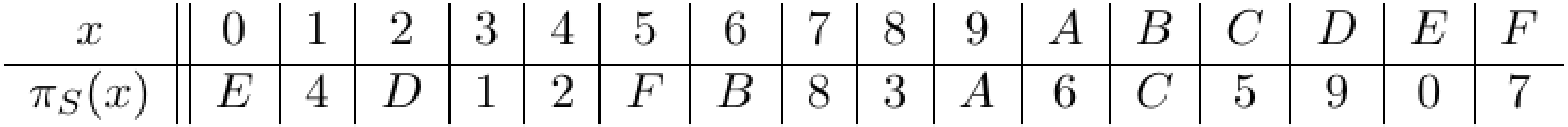
S-box Analysis: Linear Approximation table

- Suppose $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_m)$.

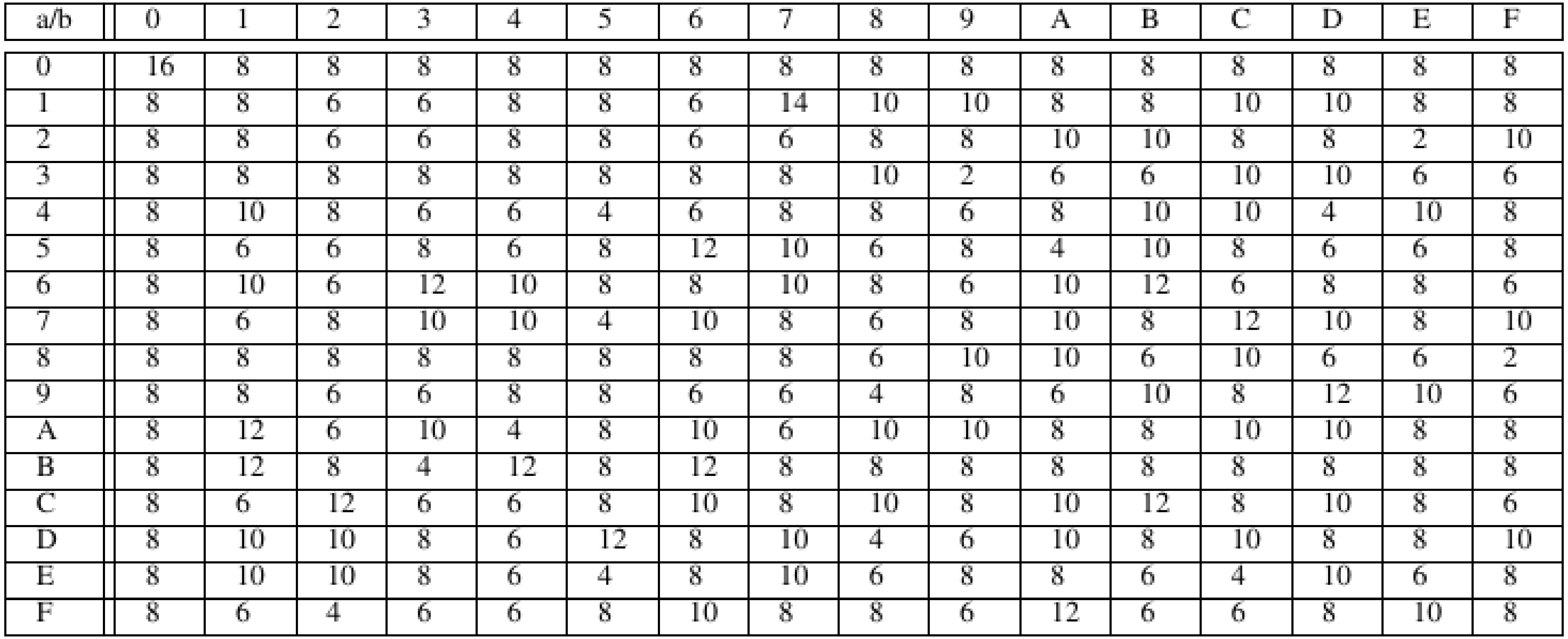
- $N_L(a, b) = \#\{ (x, y) \colon y = \pi_{S(x)}, a \cdot x \oplus b \cdot y = 0 \}$.

- For a 4×4 S-box
$$N_L(a, b) = \#\left\{ (x, y) \colon y = \pi_{S(x)}, \left(\bigoplus_{i=1}^{4} a_i x_i \right) \oplus \left(\bigoplus_{i=1}^{4} b_i y_i = 0 \right) \right\}$$

S-box Analysis: Linear Approximation Table



x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(x)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

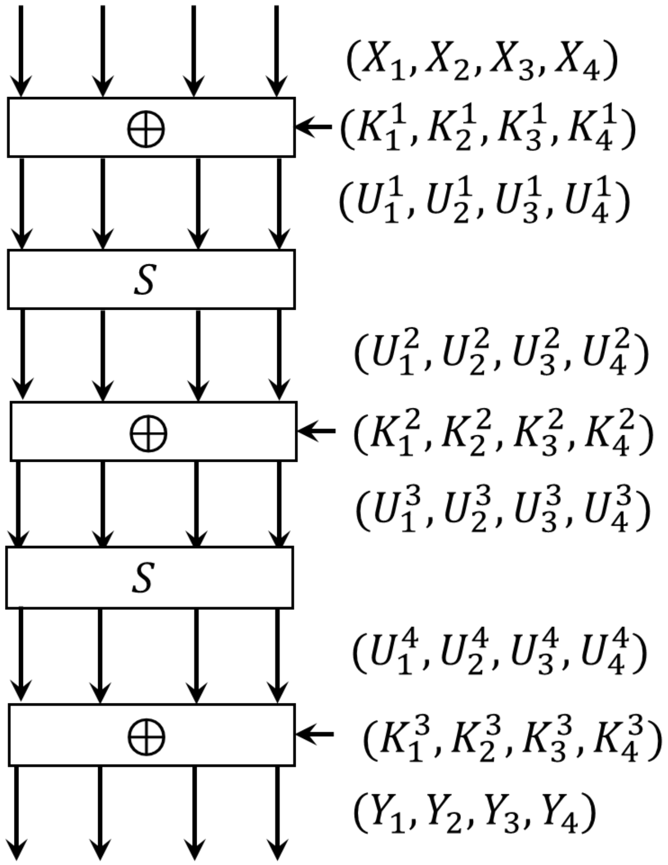


a/b	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10	12	6	8	8	6
7	8	6	8	10	10	4	10	8	6	8	10	8	12	10	8	10
8	8	8	8	8	8	8	8	8	6	10	10	6	10	6	6	2
9	8	8	6	6	8	8	6	6	4	8	6	10	8	12	10	6
A	8	12	6	10	4	8	10	6	10	10	8	8	10	10	8	8
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8	6	12	6	6	8	10	8	10	8	10	12	8	10	8	6
D	8	10	10	8	6	12	8	10	4	6	10	8	10	8	8	10
E	8	10	10	8	6	4	8	10	6	8	8	6	4	10	6	8
F	8	6	4	6	6	8	10	8	8	6	12	6	6	8	10	8

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	0111

A simple block cipher



(X_1, X_2, X_3, X_4)
$\oplus$
$(K_1^1, K_2^1, K_3^1, K_4^1)$
$(U_1^1, U_2^1, U_3^1, U_4^1)$
S
$(U_1^2, U_2^2, U_3^2, U_4^2)$
$\oplus$
$(K_1^2, K_2^2, K_3^2, K_4^2)$
$(U_1^3, U_2^3, U_3^3, U_4^3)$
S
$(U_1^4, U_2^4, U_3^4, U_4^4)$
$\oplus$
$(K_1^3, K_2^3, K_3^3, K_4^3)$
(Y_1, Y_2, Y_3, Y_4)

- $N_L(a,b) = \#\left\{ (x,y)\colon\ y = \pi_S(x), (\bigoplus_{i=1}^4 a_i x_i)\ \oplus \left(\bigoplus_{i=1}^4 b_i y_i\right) =\ 0 \right\}$
- $\epsilon(a,b) = \frac{N_L(a,b)-8}{16}$.
- From the linear approximation table, we observe:

$$N_L(1,2) = N_L(1,3) = 6$$
$$N_L(6,3) =\ N_L(A,1) = N_L(B,1) =\ N_L(B,4) =\ N_L(C,2) = 12$$

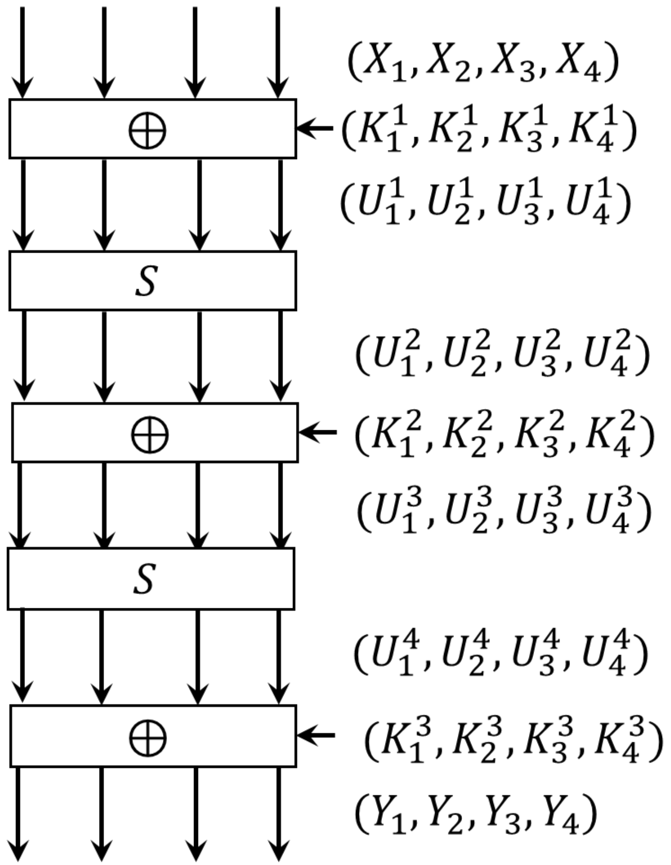
- The biases are: $\epsilon(1,2) = \epsilon(1,3) =\ -\frac{1}{8}, \epsilon(6,3) = \epsilon(A,1) =$ $\epsilon(B,1) = \epsilon(B,4) = \epsilon(C,2) = \frac{1}{4}$.
- $N_L(A,1) =\ 12$

Demonstration of the linear attack

- $\Pr[(X_2 \oplus\ X_4 \oplus\ Y_1 \oplus\ K_1^2) =\ 0] \in\ \left\{\frac{12}{16}, \frac{4}{16}\right\}$. $(N_L(A, 1) =\ 12)$
- Steps of the linear attack:
  - Suppose that we have a sample of plaintext-ciphertext pairs, $\left(x_1^j, x_2^j, x_3^j, x_4^j\right), \left(y_1^j, y_2^j, y_3^j, y_4^j\right)$, for $j\ =\ 1, \ldots, T$.
  - We assume a value of K_5, say k_5 and compute
  - $S\ =\ x_2^j \oplus\ x_4^j \oplus\ y_1^j \oplus\ k_4^2$, for $j\ =\ 1, \ldots, T$.

  - The theory tells us that if the guess of x_5 is correct then the number of times $S\ =\ x_2^j \oplus\ x_4^j \oplus\ y_1^j \oplus k_1^2$divided by T is approximately $\frac{12}{16}$ or $\frac{4}{16}$.

A slightly more complicated cipher



(X_1, X_2, X_3, X_4)
$\oplus$
$(K_1^1, K_2^1, K_3^1, K_4^1)$
$(U_1^1, U_2^1, U_3^1, U_4^1)$
S
$(U_1^2, U_2^2, U_3^2, U_4^2)$
$\oplus$
$(K_1^2, K_2^2, K_3^2, K_4^2)$
$(U_1^3, U_2^3, U_3^3, U_4^3)$
S
$(U_1^4, U_2^4, U_3^4, U_4^4)$
$\oplus$
$(K_1^3, K_2^3, K_3^3, K_4^3)$
(Y_1, Y_2, Y_3, Y_4)

- $N_L(A, 1) = 12, N_L(1,7) = 14.$
- $T_1 = U_2^1 \oplus U_4^1 \oplus U_1^2$ has bias $\frac{1}{4}$;
- $T_2 = U_1^3 \oplus U_1^4 \oplus U_2^4 \oplus U_3^4$ has bias $\frac{3}{8}$.
- Since $U_1^3 = U_1^2 \oplus K_1^2$,
- $T_1 \oplus T_2 = U_2^1 \oplus U_4^1 \oplus K_1^2 \oplus U_1^4 \oplus U_2^4 \oplus U_3^4$.
- In the next slide we consider $T_1 \oplus T_2$.

$T_1 \oplus \ T_2$

- $T_1 = U_2^1 \oplus U_4^1 \oplus U_1^2$; $\qquad U_1^3 = U_1^3 \oplus K_1^2$;
  $T_2 = U_1^3 \oplus U_1^4 \oplus U_2^4 \oplus U_3^4$

  $$T_1 = U_2^1 \oplus U_4^1 \oplus U_1^2; \quad T_2 = U_1^2 \oplus K_1^2 \oplus U_1^4 \oplus U_2^4 \oplus U_3^4$$

- $T_1 \oplus \ T_2 \ = \ U_2^1 \oplus \ U_4^1 \oplus \ K_1^2 \oplus \ U_1^4 \oplus \ U_2^4 \ \oplus \ U_3^4$

$= \ X_2 \oplus \ K_2^1 \oplus \ X_4 \oplus \ K_4^1 \oplus \ K_1^2$

$\oplus \ Y_1 \oplus \ K_1^3 \oplus \ Y_2 \oplus \ K_2^3 \oplus \ Y_3 \oplus \ K_3^3$

$= \ X_2 \oplus \ X_4 \oplus \ Y_1 \ \oplus \ Y_2 \oplus \ Y_3 \oplus \ K_1^3$

$\oplus \ K_2^3 \oplus \ K_3^3 \oplus \ K_2^1 \oplus \ K_4^1 \oplus \ K_1^2.$

$T_1 \oplus T_2$

- $bias(T_1) = \frac{1}{4}, bias(T_2) = \frac{3}{8}.$

- By Piling-up lemma

$$bias(T_1 \oplus T_2)$$

$$= 2 \times \frac{1}{4} \times \frac{3}{8} = \frac{3}{16}.$$

Steps of Linear Attack

- Suppose that we have a sample of planetext-ciphertext pairs, $\left(x_1^j, x_2^j, x_3^j, x_4^j\right), \left(y_1^j, y_2^j, y_3^j, y_4^j\right)$, for$j = 1, \dots, T$.

- We assume a value of k_1^3, k_2^3, k_3^3, and compute

$$= x_2^j x_4^j \oplus y_1^j \oplus y_2^j \oplus y_3^j \oplus k_1^3 \oplus k_2^3 \oplus k_3^3, for\ j = 1, \dots, T.$$

- The theory says that if the guess of x_5 is correct then the number of times $S = x_2^j \oplus x_4^j \oplus y_1^j \oplus k_5$ divided by T is approximately $\frac{11}{16}$ or $\frac{5}{16}$.