Cryptanalysis of Block Ciphers

Dept. of Computer Science & Engg.
IIT Kharagpur, India

Dipanwita Roy Chowdhury

Types of Cryptanalysis/Attacks

- Algebraic Analysis

  Linear Cryptanalysis, Differential Cryptanalysis

- Algorithmic / Structural Analysis

  Man-in-the-Middle Attack, Related Key Attack

- Side Channel Analysis

  Power Attack, Timing Attack, Fault Analysis etc.

Algebraic Analysis/Attacks

- Introduction
- Substitution-Permutation Networks
- Linear cryptanalysis
- Differential cryptanalysis

Introduction

- **A commonly used design for modern-day block ciphers is that of an iterated cipher:**
  - The cipher requires the specification of a **round function** and a **key schedule**, and the encryption of a plaintext will proceed through Nr similar *rounds*.
    - **random key K:** used to construct Nr **round keys** (also called **subkeys**), which are denoted $K^1,\ldots,K^{Nr}$.
    - **key schedule ($K^1,\ldots,K^{Nr}$):** constructed from K using a fixed, public algorithm.
    - **round function g:** takes two inputs: a round key (K^r) and a current **state** (w^{r-1}). $w^r=g(w^{r-1},K^r)$ is the next state.
    - **plaintext x:** the initial state w^0.
    - **Ciphertext y:** the state after all Nr rounds done.

Introduction

- **Encryption operations:**

$$
\begin{aligned}
w^0 &\leftarrow x \\
w^1 &\leftarrow g(w^0, K^1) \\
w^2 &\leftarrow g(w^1, K^2) \\
&\vdots \\
w^{Nr-1} &\leftarrow g(w^{Nr-2}, K^{Nr-1}) \\
w^{Nr} &\leftarrow g(w^{Nr-1}, K^{Nr}) \\
y &\leftarrow w^{Nr}
\end{aligned}
$$

Decryption operations:

$$
\begin{aligned}
w^{Nr} &\leftarrow y \\
w^{Nr-1} &\leftarrow g^{-1}(w^{Nr}, K^{Nr}) \\
&\vdots \\
w^1 &\leftarrow g^{-1}(w^2, K^2) \\
w^0 &\leftarrow g^{-1}(w^1, K^1) \\
x &\leftarrow w^0
\end{aligned}
$$

Note: function g is injective (one-to-one)

Substitution-Permutation Networks (SPN)

- **Cryptosystem : *SPN***
  - l, m and Nr are positive integers
  - $\pi_S : \{0,1\}^l \rightarrow \{0,1\}^l$ is a permutation
  - $\pi_P : \{1,\ldots, lm\} \rightarrow \{1,\ldots, lm\}$ is a permutation.
  - $P = C = \{0,1\}^{lm}$, and $K \subseteq (\{0,1\}^{lm})^{Nr+1}$ consist 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 .

Substitution-Permutation Networks

- **Algorithm : SPN** $(x, \pi_S, \pi_P, (K^1, \ldots, K^{Nr+1}))$

$w^0 \leftarrow x$

for $r \leftarrow 1$ to $Nr-1$

do
- $u^r \leftarrow w^{r-1} \oplus K^r$
- for $i \leftarrow 1$ to m
  - do $v_{(i)}^r \leftarrow \pi_S(u_{(i)}^r)$
- $w^r \leftarrow (v_{\pi_P(1)}^r, \ldots, v_{\pi_P(lm)}^r)$

$u^{Nr} \leftarrow w^{Nr-1} \oplus K^{Nr}$

for $i \leftarrow 1$ to m

do $v_{(i)}^{Nr} \leftarrow \pi_S(u_{(i)}^{Nr})$

$y \leftarrow v^{Nr} \oplus K^{Nr+1}$

output(y)

u^r : input to the S-boxes in round r.
v^r : output of the S-boxes in round r.
w^r : obtained from v^r by applying π_p .
u^{r+1} : constructed from w^r by xor-ing with the round key K^{r+1} (called round key mixing).

Substitution-Permutation Networks

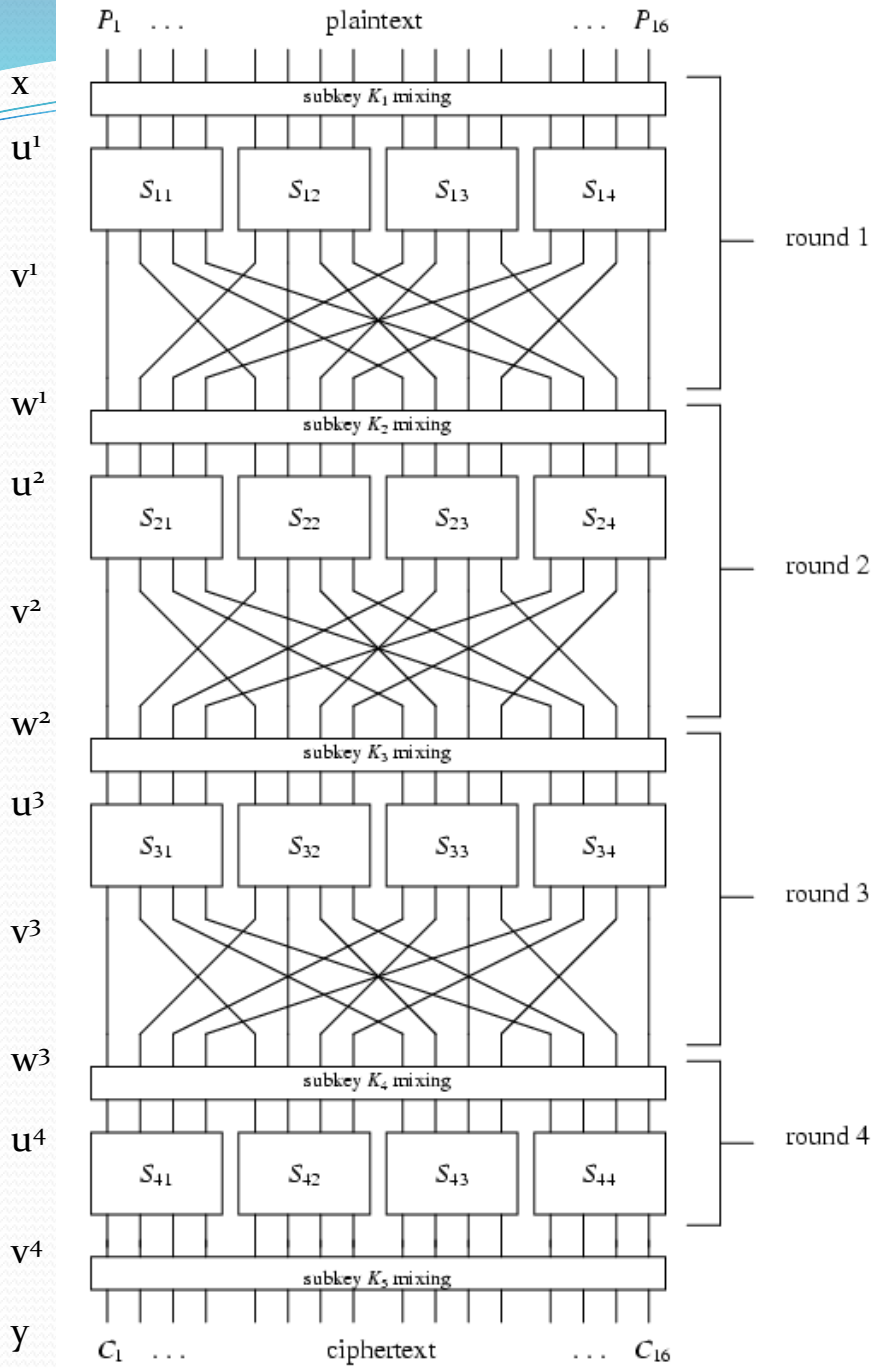
- **Example :**
  - Suppose $l = m = Nr = 4$.
  - Let π_S be defined as follows, where the input and the output are written in hexadecimal: e.g. input to Sbox: 0101 Output: 1111

input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
output	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Let π_P be defined as follows: output line 5 ids permuted to 2

input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
output	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

See **Figure** for a pictorial representation of this particular SPN, where S_{ir} means i-th round, r-th S-box.



A substitution-permutation network

Substitution-Permutation Networks

- **Key schedule:** suppose we begin with a 32-bit key $K = (k_1, \ldots, k_{32}) \in \{0,1\}^{32}$. For $1 \le r \le 5$, define K^r to consist of 16 consecutive bits of K, beginning with k_{4r-3}.
- K= 0011 1010 1001 0100 1101 0110 0011 1111
- Round keys:

  K^1= 0011 1010 1001 0100

  K^2= 1010 1001 0100 1101

  K^3= 1001 0100 1101 0110

  K^4= 0100 1101 0110 0011

  K^5= 1101 0110 0011 1111

Substitution-Permutation Networks

- Suppose the plaintext is x= 0010 0110 1011 0111.
- Then the encryption of x proceeds as follows:

w^0= 0010 0110 1011 0111

K^1= 0011 1010 1001 0100

u^1= 0001 1100 0010 0011

v^1= 0100 0101 1101 0001

w^1= 0010 1110 0000 0111

K^2= 1010 1001 0100 1101

u^2= 1000 0111 0100 1010

v^2= 0011 1000 0010 0110

w^2= 0100 0001 1011 1000

Substitution-Permutation Networks

K^3= 1001 0100 1101 0110

u^3= 1101 0101 0110 1110

v^3= 1001 1111 1011 0000

w^3= 1110 0100 0110 1110

K^4= 0100 1101 0110 0011

u^4= 1010 1001 0000 1101

v^4= 0110 1010 1110 1001

K^5= 1101 0110 0011 1111, and

y= 1011 1100 1101 0110

is the ciphertext.

□

Linear Cryptanalysis

- We want to find a probability of linear relationship between a subset of plaintext bits and a subset of data bits preceding the last round. This relation behaves in a non-random fashion.
- **known plaintext attack :** The attacker has a lot of plaintext-ciphertext pairs.
- For each candidate subkey, we partially decrypt the cipher and check if the relation holds. If the relation holds then increment its corresponding counter. At the end, the candidate key that counts furthest from ½ is the most likely subkey.

Linear Cryptanalysis

- **The Piling-up Lemma**
  - Suppose $X_1, X_2, \ldots$ are independent random variables from {0,1}. And

$$\Pr[X_i = 0] = p_i, \quad i = 1,2,\ldots \text{ Hence,}$$

$$\Pr[X_i = 1] = 1 - p_i, \quad i = 1,2,\ldots$$

  - The independence of X_i, X_j implies

$$\Pr[X_i = 0, X_j = 0] = p_i p_j$$

$$\Pr[X_i = 0, X_j = 1] = p_i(1 - p_j)$$

$$\Pr[X_i = 1, X_j = 0] = (1 - p_i) p_j$$

$$\Pr[X_i = 1, X_j = 1] = (1 - p_i)(1 - p_j)$$

Linear Cryptanalysis

- Now consider $X_i \oplus X_j$

$$\Pr[X_i \oplus X_j = 0] = p_i p_j + (1-p_i)(1-p_j)$$

$$\Pr[X_i \oplus X_j = 1] = p_i (1-p_j) + (1-p_i) p_j$$

- The **bias** of X_i is defined to be the quantity

$$\varepsilon_i = p_i - \tfrac{1}{2}$$

- And we have

$$-\tfrac{1}{2} \le \varepsilon_i \le \tfrac{1}{2},$$

$$\Pr[X_i = 0] = \tfrac{1}{2} + \varepsilon_i,$$

$$\Pr[X_i = 1] = \tfrac{1}{2} - \varepsilon_i.$$

Linear Cryptanalysis

- Let $\varepsilon_{i_1,i_2,\ldots,i_k}$ denote the bias of $X_{i_1} \oplus \cdots \oplus X_{i_k}$

- **Piling-up lemma:** Let $\varepsilon_{i_1,i_2,\ldots,i_k}$ denote the bias of the random variable $X_{i_1} \oplus \cdots \oplus X_{i_k}$.

  Then

$$\varepsilon_{i_1,i_2,\ldots,i_k} = 2^{k-1} \prod_{j=1}^{k} \varepsilon_{i_j} .$$

Corollary: Let $\varepsilon_{i_1,i_2,\ldots,i_k}$ denote the bias of the random variable $X_{i_1} \oplus \cdots \oplus X_{i_k}$.

Suppose that $\varepsilon_{i_j} = 0$ for some j. Then $\varepsilon_{i_1,i_2,\ldots,i_k} = 0$.

Linear Cryptanalysis

- **Linear Approximations of S-boxes**
  - Consider an S-box $\pi_S : \{0,1\}^m \rightarrow \{0,1\}^n$
  - Let the input m-tuple be $X=(x_1,\ldots,x_m)$. And the output n-tuple be $Y=(y_1,\ldots,y_n)$.
  - We can see that

$$\Pr[X_1 = x_1,\ldots, X_m = x_m, Y_1 = y_1,\ldots, Y_n = y_n] = 0$$

if $(y_1,\ldots, y_n) \neq \pi_S(x_1,\ldots, x_m)$; and

$$\Pr[X_1 = x_1,\ldots, X_m = x_m, Y_1 = y_1,\ldots, Y_n = y_n] = 2^{-m}$$

if $(y_1,\ldots, y_n) = \pi_S(x_1,\ldots, x_m)$.

  - Now we can compute the bias of the form

$$X_{i_1} \oplus \cdots \oplus X_{i_k} \oplus Y_{j_1} \oplus \cdots \oplus Y_{j_l}$$

Linear Cryptanalysis

- **Example** : We use the S-box as **Example**.

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	$X_2 \oplus X_3$	$Y_1 \oplus Y_3 \oplus Y_4$	$X_1 \oplus X_4$	Y_2	$X_3 \oplus X_4$	$Y_1 \oplus Y_4$
0	0	0	0	1	1	1	0	0	0	0	1	0	1
0	0	0	1	0	1	0	0	0	0	1	1	1	0
0	0	1	0	1	1	0	1	1	0	0	1	1	0
0	0	1	1	0	0	0	1	1	1	1	0	0	1
0	1	0	0	0	0	1	0	1	1	0	0	0	0
0	1	0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	0	1	0	1	1	0	1	0	0	1	0
0	1	1	1	1	0	0	0	0	1	1	0	0	1
1	0	0	0	0	0	1	1	0	0	1	0	0	1
1	0	0	1	1	0	1	0	0	0	0	0	1	1
1	0	1	0	0	1	1	0	1	1	1	1	1	0
1	0	1	1	1	1	0	0	1	1	0	1	0	1
1	1	0	0	0	1	0	1	1	1	1	1	0	1
1	1	0	1	1	0	0	1	1	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1	0	1	0
1	1	1	1	0	1	1	1	0	0	0	1	0	1

Linear Cryptanalysis

- Consider $X_1 \oplus X_4 \oplus Y_2$. The probability that $X_1 \oplus X_4 \oplus Y_2 = 0$ can be determined by counting the number of rows in which $X_1 \oplus X_4 \oplus Y_2 = 0$ and then dividing by 16.
- It is seen that

$$\Pr[X_1 \oplus X_4 \oplus Y_2 = 0] = \frac{1}{2}$$

  Hence, the bias is 0.
- If we instead analyze $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4$, we find that the bias is –3/8.

□

Linear Cryptanalysis

- We can record the bias of all 2^8=256 possible random variables.
- We represent the relevant random variable in the form

  where $\left(\bigoplus_{i=1}^{4} a_i X_i\right) \oplus \left(\bigoplus_{i=1}^{4} b_i Y_i\right)$.

  $$a_i \in \{0,1\}, b_i \in \{0,1\}, i = 1,2,3,4$$

- We treat (a_1,a_2,a_3,a_4) and (b_1,b_2,b_3,b_4) as hexadecimal digit (they are called **input sum** and **output sum**, respectively)

Linear Cryptanalysis

- Let $N_L(a,b)$ denote the number of binary eight-tuples $(x_1,x_2,x_3,x_4,y_1,y_2,y_3,y_4)$ s.t

$$(y_1, y_2, y_3, y_4) = \pi_S(x_1, x_2, x_3, x_4)$$

and

$$\left(\bigoplus_{i=1}^{4} a_i X_i\right) \oplus \left(\bigoplus_{i=1}^{4} b_i Y_i\right) = 0$$

The bias is computed as $\varepsilon(a,b) = (N_L(a,b) - 8)/16$

- The table of all N_L is called the **linear approximation table** .

Input Sum	Output Sum 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	−2	−2	0	0	−2	+6	+2	+2	0	0	+2	+2	0	0
2	0	0	−2	−2	0	0	−2	−2	0	0	+2	+2	0	0	−6	+2
3	0	0	0	0	0	0	0	0	+2	−6	−2	−2	+2	+2	−2	−2
4	0	+2	0	−2	−2	−4	−2	0	0	−2	0	+2	+2	−4	+2	0
5	0	−2	−2	0	−2	0	+4	+2	−2	0	−4	+2	0	−2	−2	0
6	0	+2	−2	+4	+2	0	0	+2	0	−2	+2	+4	−2	0	0	−2
7	0	−2	0	+2	+2	−4	+2	0	−2	0	+2	0	+4	+2	0	+2
8	0	0	0	0	0	0	0	0	−2	+2	+2	−2	+2	−2	−2	−6
9	0	0	−2	−2	0	0	−2	−2	−4	0	−2	+2	0	+4	+2	−2
A	0	+4	−2	+2	−4	0	+2	−2	+2	+2	0	0	+2	+2	0	0
B	0	+4	0	−4	+4	0	+4	0	0	0	0	0	0	0	0	0
C	0	−2	+4	−2	−2	0	+2	0	+2	0	+2	+4	0	+2	0	−2
D	0	+2	+2	0	−2	+4	0	+2	−4	−2	+2	0	+2	0	0	+2
E	0	+2	+2	0	−2	−4	0	+2	−2	0	0	−2	−4	+2	−2	0
F	0	−2	−4	−2	−2	0	+2	0	0	−2	+4	−2	−2	0	+2	0

Figure: Linear approximation table values of $N_L(a,b)-8$

Linear Cryptanalysis

- **Linear Attack on an SPN**
  - Linear cryptanalysis requires a set of linear approximations of S-boxes that can be used to derive a linear approximation of the entire SPN (excluding the last round).
  - **Figure** illustrates the structure of the approximation we will use.
    - Arrows are the random variables involved in the approximations and the labeled S-boxes (**active S-boxes**) are used in the approximations.

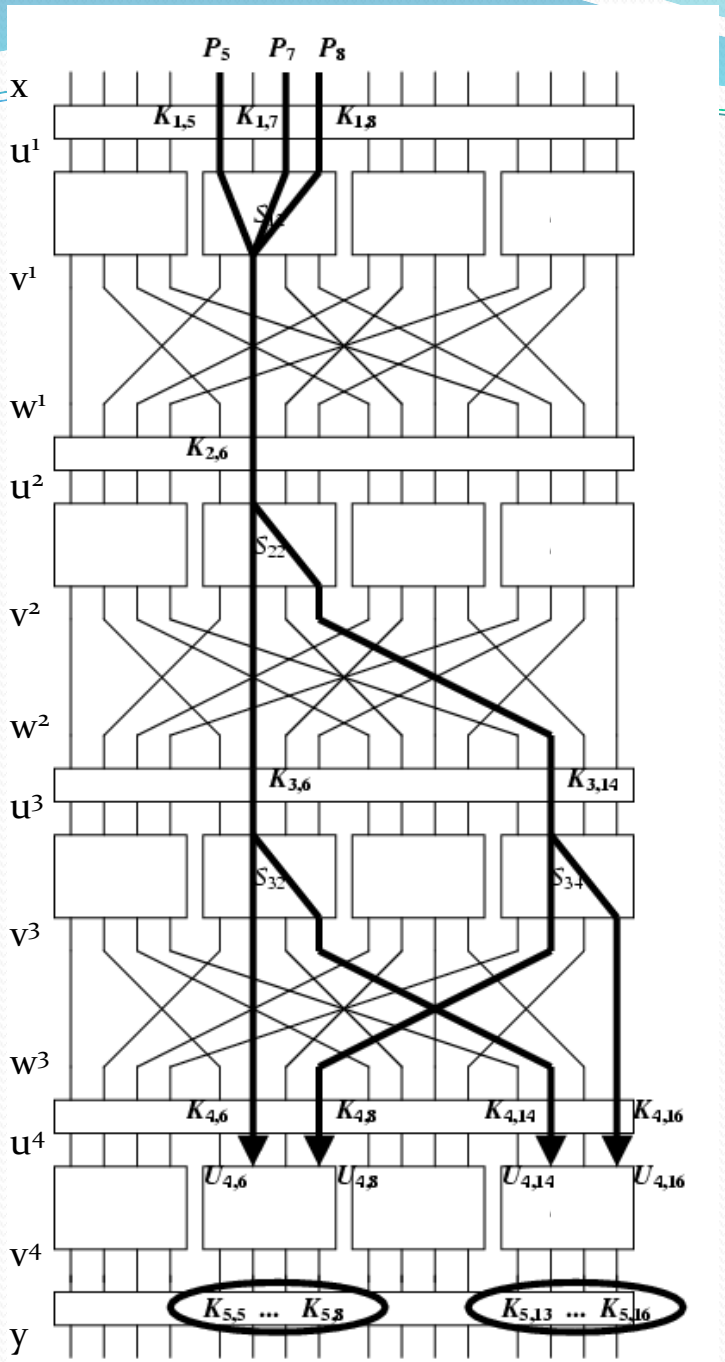


Figure: A linear approximation of an SPN

Linear Cryptanalysis

- The approximation incorporates four active S-boxes:
  - In S_{12}, $T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1$ has bias ¼
  - In S_{22}, $T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2$ has bias -¼
  - In S_{32}, $T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3$ has bias -¼
  - In S_{34}, $T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$ has bias -¼
- T_1, T_2, T_3, T_4 have biases that are high in absolute value. Further, we will see their XOR will lead to **cancellations of "intermediate"** random variables.

Linear Cryptanalysis

- Using **Piling-up lemma**, $T_1 \oplus T_2 \oplus T_3 \oplus T_4$ has bias equal to $2^3(1/4)(-1/4)^3=-1/32$.
  - Note: we assume the four r.v are independent.
- Then T_1, T_2, T_3, T_4 can be expressed in terms of plaintext bits, bits of u^4 (input to the last round) and key bits as follows:

$$T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1 = X_5 \oplus K_5^1 \oplus X_7 \oplus K_7^1 \oplus X_8 \oplus K_8^1 \oplus V_6^1$$
$$T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2 = V_6^1 \oplus K_6^2 \oplus V_6^2 \oplus V_8^2$$
$$T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3 = V_6^2 \oplus K_6^3 \oplus V_6^3 \oplus V_8^3$$
$$T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3 = V_8^2 \oplus K_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$$

Linear Cryptanalysis

- XOR the right side and we get

$$X_5 \oplus X_7 \oplus X_8 \oplus V_6^3 \oplus V_8^3 \oplus V_{14}^3 \oplus V_{16}^3 \\ \oplus K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3 \qquad (3.1)$$

- Then replace V_i^3 by U_i^4 and key bits:

$$V_6^3 = U_6^4 \oplus K_6^4 \qquad V_8^3 = U_{14}^4 \oplus K_{14}^4$$
$$V_{14}^3 = U_8^4 \oplus K_8^4 \qquad V_{16}^3 = U_{16}^4 \oplus K_{16}^4$$

- Now substitute them into 3.1:

$$X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 \\ \oplus K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3 \oplus K_6^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4 \quad (3.2)$$

Linear Cryptanalysis

- The expression above only involves plaintext bits, bits of u^4 and key bits.
- Suppose the key bits are fixed. Then

$$K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3 \oplus K_6^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4$$

  has the (fixed) value 0 or 1.

- It follows that

$$X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 \qquad (3.3)$$

  has bias -1/32 or 1/32 where the sign depends on the key bits (=0 or =1).

Linear Cryptanalysis

- The fact that (3.3) has bias bounded away from o allows us to carry out linear attack.
- Suppose that we have T plaintext-ciphertext pairs (denoted by τ), all use the same unknown key, K. The attack will allow us to obtain the eight key bits,

$$K_5^5, K_6^5, K_7^5, K_8^5, K_{13}^5, K_{14}^5, K_{15}^5, K_{16}^5$$

- There are 2^8=256 possibilities for the eight key bits. We refer to a binary 8-tuple as a **candidate subkey**.

Linear Cryptanalysis

- For each $(x, y) \in \tau$ and for each candidate subkey, we compute a partial decryption of y and obtain the resulting value for $u^4_{(2)}, u^4_{(4)}$.
- Then we compute the value

$$x_5 \oplus x_7 \oplus x_8 \oplus u^4_6 \oplus u^4_8 \oplus u^4_{14} \oplus u^4_{16} \qquad (3.4)$$

- We maintain an array of counters indexed by the 256 possible candidate subkeys, and increment the counter corresponding to a particular subkey when (3.4) has the value 0.
- In the end, we expect most counters will have a value close to T/2, but the correct candidate subkey will close to T/2±T/32.

Linear Cryptanalysis

- The attack is presented as **Algorithm**.
  - L_1 and L_2 are hexadecimal value.
  - π_S^{-1} is the inverse of the S-box.
  - The output, maxkey, contains the most likely subkey.
- In general, it is suggested that a linear attack based on a linear approximation having bias ε will be successful if the number of plaintext-ciphertext pairs is approximately $c\varepsilon^{-2}$ for some "small" constant c.

Algorithm: $\text{LINEARATTACK}(\tau, T, \pi_S^{-1})$

for $(L_1, L_2) \leftarrow (0,0)$ to (F, F)
  do $Count[L_1, L_2] \leftarrow 0$
for each $(x, y) \in \tau$
do
  for $(L_1, L_2) \leftarrow (0,0)$ to (F, F)
  do
    $v^4_{(2)} \leftarrow L_1 \oplus y_{(2)}$
    $v^4_{(4)} \leftarrow L_2 \oplus y_{(4)}$
    $u^4_{(2)} \leftarrow \pi_S^{-1}(v^4_{(2)})$
    $u^4_{(4)} \leftarrow \pi_S^{-1}(v^4_{(4)})$
    $z \leftarrow x_5 \oplus x_7 \oplus x_8 \oplus u^4_6 \oplus u^4_8 \oplus u^4_{14} \oplus u^4_{16}$
    if $z = 0$
      then $Count[L_1, L_2] \leftarrow Count[L_1, L_2] + 1$
$max \leftarrow -1$
for $(L_1, L_2) \leftarrow (0,0)$ to (F, F)
  do
    $Count[L_1, L_2] \leftarrow |Count[L_1, L_2] - T/2|$
    if $Count[L_1, L_2] > \max$
      then
        $max \leftarrow Count[L_1, L_2]$
        $maxkey \leftarrow (L_1, L_2)$
output $(maxkey)$

Differential Cryptanalysis

- The main difference from linear attack is that differential attack involves comparing the XOR of two inputs to the XOR of the corresponding outputs.
- Differential attack is a **chosen-plaintext attack**.
- We consider inputs x and x* having a specified XOR value denoted by $x' = x \oplus x^*$
- We decrypt y and y* using all possible key and determine if their XOR has a certain value. Whenever it does, increment the corresponding counter. At the end, we expect the largest one is the most likely subkey.

Differential Cryptanalysis

- **Definition:**
  - Let $\pi_S : \{0,1\}^m \rightarrow \{0,1\}^n$ be an S-box. Consider an (ordered) pair of bitstrings of length m, say (x,x*). We say that the input XOR of the S-box is $x \oplus x*$ and the output XOR is $\pi_S(x) \oplus \pi_S(x*)$

  For any, $x' \in \{0,1\}^m$ define the set $\Delta(x')$ to consist of all the ordered pairs (x,x*) having input XOR equal to x'.

Differential Cryptanalysis

- It is easy to see that any set $\Delta(x')$ contains 2^m pairs, and that

$$\Delta(x') = \{(x, x \oplus x') : x \in \{0,1\}^m\}$$

- For each pair in $\Delta(x')$ we can compute the output XOR of the S-box. Then we can tabulate the distribution of output XORs. There are 2^m output XORs which are distributed among 2^n possible values.
  - A non-uniform output distribution will be the basis for a successful attack.

Differential Cryptanalysis

- **Example:**
  - We use the same S-box as before. Suppose we consider input XOR x'=1011. Then

$$\Delta(1011) = \{(0000,1011), (0001,1010), \ldots, (1111,0100)\}$$

  - We compute the following table, where

$$x \oplus x^* = 1011,$$

$$y = \pi_S(x), y^* = \pi_S(x^*),$$

$$y' = y \oplus y^*$$

x	x*	y	y*	y’
0000	1011	1110	1100	0010
0001	1010	0100	0110	0010
0010	1001	1101	1010	0111
0011	1000	0001	0011	0010
0100	1111	0010	0111	0101
0101	1110	1111	0000	1111
0110	1101	1011	1001	0010
0111	1100	1000	0101	1101
1000	0011	0011	0001	0010
1001	0010	1010	1101	0111
1010	0001	0110	0100	0010
1011	0000	1100	1110	0010
1100	0111	0101	1000	1101
1101	0110	1001	1011	0010
1110	0101	0000	1111	1111
1111	0100	0111	0010	0101

0000	0	1000	0
0001	0	1001	0
0010	8	1010	0
0011	0	1011	0
0100	0	1100	0
0101	2	1101	2
0110	0	1110	0
0111	2	1111	2

Number of output

Distribution table for x’=1011

Differential Cryptanalysis

- In **Example**, only 5 of the 16 possible output XORs occur. It has a very non-uniform distribution.
- We can compute all possible input XORs as **Example**.
- Define

$$N_D(x', y') = |\{(x, x^*) \in \Delta(x') : \pi_S(x) \oplus \pi_S(x^*) = y'\}|$$

  - $N_D(x',y')$ counts the number of pairs with input XOR equal to x' and output XOR equal to y'. (**Figure**)

Input Difference	Output Difference 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Example

Figure: Difference distribution table: values of $N_D(x',y')$

Differential Cryptanalysis

- An input XOR is computed as

$$u_{(i)}^{r} \oplus (u_{(i)}^{r})^{*} = (w_{(i)}^{r-1} \oplus K_{(i)}^{r}) \oplus ((w_{(i)}^{r-1})^{*} \oplus K_{(i)}^{r})$$

$$= w_{(i)}^{r-1} \oplus (w_{(i)}^{r-1})^{*}$$

  - Therefore, the input XOR does not depend on the subkey bits used in round r; it is equal to the (permuted) output XOR of round r-1.

- Let a' denote the input XOR and let b' denote the output XOR. (a',b') is called a **differential**.

Differential Cryptanalysis

- **propagation ratio $R_p(a', b')$**:

$$R_p(a', b') = \frac{N_D(a', b')}{2^m}$$

  - $R_p(a', b')$ can be interpreted as a conditional probability:

$$\Pr[\text{output XOR} = b' \mid \text{input XOR} = a'] = R_p(a', b')$$

- We combine differentials in consecutive rounds to form a **differential trail**. A particular differential trail is shown in **Figure**.

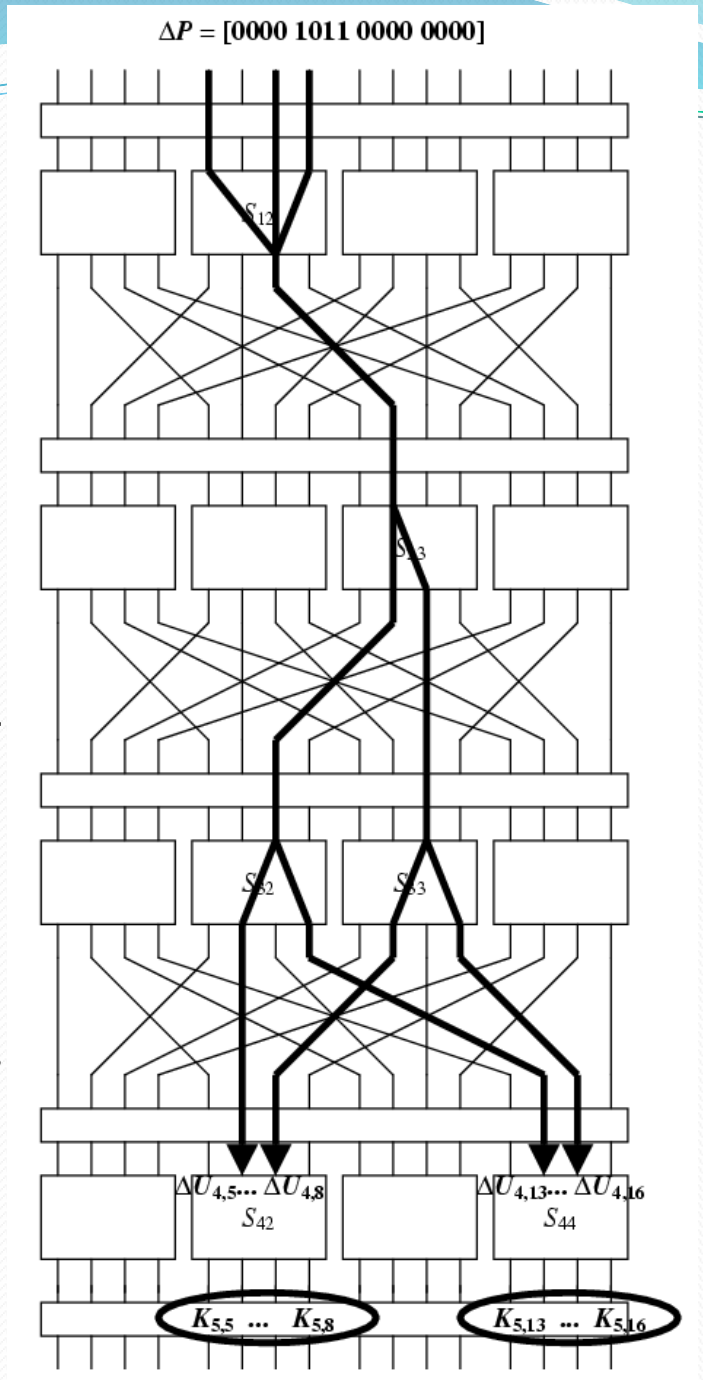


Figure: A differential trail for a SPN

Differential Cryptanalysis

- The differential attack arising from **Figure** uses the following propagation ratios of differentials:
  - In S_{12}, $R_p(1011,0010) = 1/2$
  - In S_{23}, $R_p(0100,0110) = 3/8$
  - In S_{32}, $R_p(0010,0101) = 3/8$
  - In S_{33}, $R_p(0010,0101) = 3/8$
- We therefore obtain a propagation ratio for a differential trail of the first three rounds of the SPN:

$$R_p(0000\ 1011\ 0000\ 0000, 0000\ 0101\ 0101\ 0000) = \frac{1}{2} \times \left(\frac{3}{8}\right)^3 = \frac{27}{1024}$$

Differential Cryptanalysis

- In other words,

$$x' = 0000\ 1011\ 0000\ 0000 \Rightarrow (v^3)' = 0000\ 0101\ 0101\ 0000$$

with probability 27/1024. However,

$$(v^3)' = 0000\ 0101\ 0101\ 0000 \Leftrightarrow (u^4)' = 0000\ 0110\ 0000\ 0110$$

Hence, it follows that

$$x' = 0000\ 1011\ 0000\ 0000 \Rightarrow (u^4)' = 0000\ 0110\ 0000\ 0110$$

with probability 27/1024.

Differential Cryptanalysis

- **Algorithm** presents the attack algorithm.
- The input and output are similar to linear attack, except that τ is a set (x,x*,y,y*), where x' is fixed.
- **Algorithm** makes use of a certain **filtering operation**. Tuples (x,x*,y,y*) for which the differential holds are often called **right pairs**, and allow us to determine the key bits.

  A right pair has the form $(u^4_{(1)})' = (u^4_{(3)})' = 0000$

  Hence we consider those $y_{(1)} = (y_{(1)})^*$ and $y_{(3)} = (y_{(3)})^*$.

Algorithm : DIFFERENTIALATTACK(τ, T, π_S^{-1})

for $(L_1, L_2) \leftarrow (0,0)$ to (F, F)
do $Count[L_1, L_2] \leftarrow 0$
for each $(x, y, x^*, y^*) \in \tau$
do
  if $(y_{(1)} = (y_{(1)})^*)$ and $(y_{(3)} = (y_{(3)})^*)$
  then
    for $(L_1, L_2) \leftarrow (0,0)$ to (F, F)
    do
      $v^4_{(2)} \leftarrow L_1 \oplus y_{(2)}$
      $v^4_{(4)} \leftarrow L_2 \oplus y_{(4)}$
      $u^4_{(2)} \leftarrow \pi_S^{-1}(v^4_{(2)})$
      $u^4_{(4)} \leftarrow \pi_S^{-1}(v^4_{(4)})$
      $(v^4_{(2)})^* \leftarrow L_1 \oplus (y_{(2)})^*$
      $(v^4_{(4)})^* \leftarrow L_2 \oplus (y_{(4)})^*$
      $(u^4_{(2)})^* \leftarrow \pi_S^{-1}((v^4_{(2)})^*)$
      $(u^4_{(4)})^* \leftarrow \pi_S^{-1}((v^4_{(4)})^*)$
      $(u^4_{(2)})' \leftarrow u^4_{(2)} \oplus (u^4_{(2)})^*$
      $(u^4_{(4)})' \leftarrow u^4_{(4)} \oplus (u^4_{(4)})^*$
      if $((u^4_{(2)})' = 0110)$ and $((u^4_{(4)})' = 0110)$
        then $Count[L_1, L_2] \leftarrow Count[L_1, L_2] + 1$

$max \leftarrow -1$
for $(L_1, L_2) \leftarrow (0,0)$ to (F, F)
do
  if $Count[L_1, L_2] > max$
  then
    $max \leftarrow Count[L_1, L_2]$
    $maxkey \leftarrow (L_1, L_2)$
output $(maxkey)$

Differential Cryptanalysis

- A differential attack based on a differential trail having propagation ratio equal to ε will often be successful if the number of tuples (x,x*,y,y*), which we denote by T, is approximately $c\varepsilon^{-1}$, for a "small" constant c.

References

- **Cryptography: Theory and Practice,** Douglas R. Stinson

  November 1, 2005 by Chapman and Hall/CRC, Textbook - 616

  Pages – 27 B/W Illustrations, ISBN 9781584885085 - CAT# C5084
  Series: Discrete Mathematics and Its Applications

- **A Tutorial on Linear and Differential Cryptanalysis**

  https://www.engr.mun.ca/~howard/PAPERS/ldc_**tutorial**.pdf