Cryptanalysis

Jiageng Chen

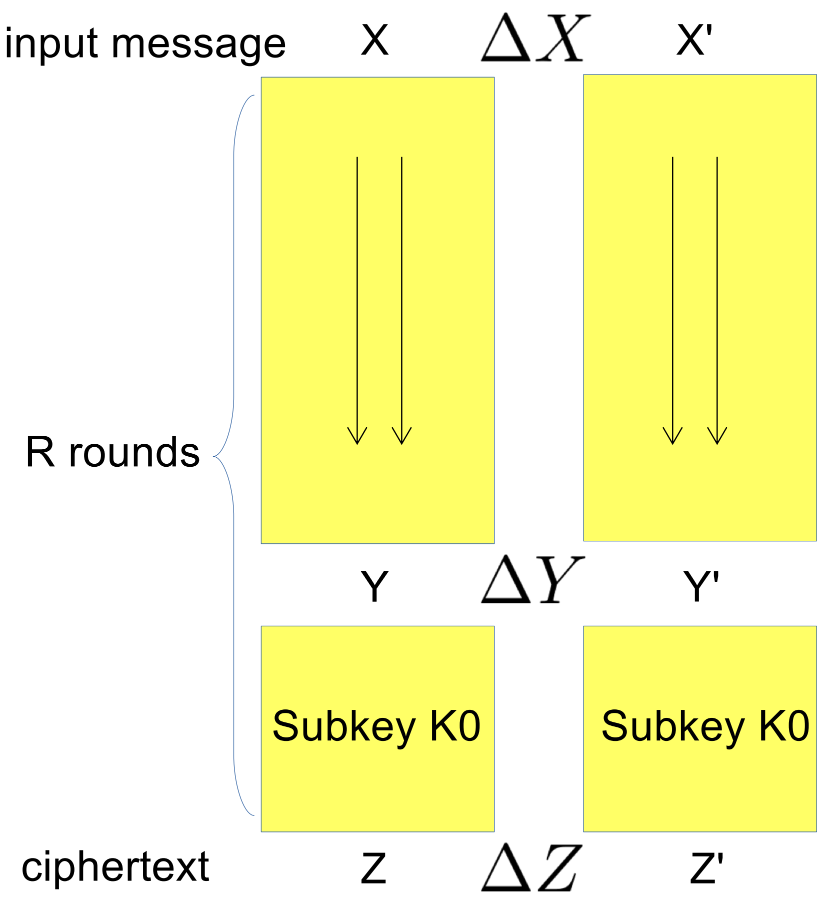
Content

- Overview
- Block Ciphers:
    - Linear
    - Differential
    - Other Attacks
    - Statistical Analysis

Differential and Linear Cryptanalysis Origins

- Differential cryptanalysis originally defined on DES
- Eli Biham and Adi Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer Verlag, 1993.
- Linear cryptanalysis first defined on Feal by Matsui and Yamagishi, 1992.
- Matsui later published a linear attack on DES.

Differential Cryptanalysis



1. Block ciphers are usually composed by iterating R rounds of similar nonlinear operations.

2. We track the difference value of input messages X to Y, try to build an efficient distinguisher

3. Then the attacker by guessing subkey K0 used in last rounds, decrypt Z to match Y.

4. The statistical behavior for the correct key K0 will be much more significant than other wrong keys, which allow us to identify the correct the key k0.

5. The rest of the subkey can be recovered in the same way by peeling off last rounds.

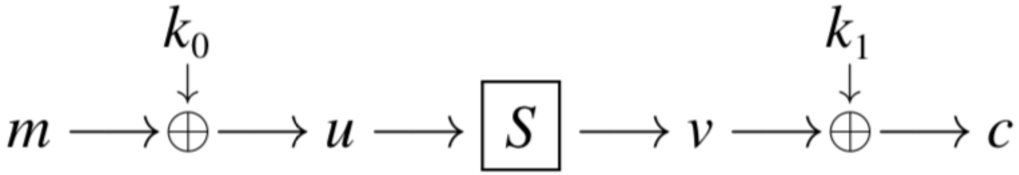
Efficient long differential path $\Delta X \rightarrow \Delta Y$ is crucial to the success of the attack

Differential Cryptanalysis - Simple case

- Consider the simple XOR encryption : $c = m \oplus k$
- What if we use the key twice?
    - $c_0 \oplus c_1 = (m_0 \oplus k) \oplus (m_1 \oplus k) = m_0 \oplus m_1$
- While we might not get much information from considering a single message and ciphertext, we might gain much more by considering pairs of messages and ciphertext
- Secret key k could be entirely removed by simply manipulating the ciphertexts

Cipher One

CIPHERONE$(m_0, k_0 \| k_1)$	CIPHERONE$(m_1, k_0 \| k_1)$
$u_0 = m_0 \oplus k_0$ $v_0 = S[u_0]$ $c_0 = v_0 \oplus k_1$	$u_1 = m_1 \oplus k_0$ $v_1 = S[u_1]$ $c_1 = v_1 \oplus k_1$



- Trace a difference between two plaintexts
- Cryptanalyst does know the value of the difference between these two internal values since

$$u_0 \oplus u_1 = (m_0 \oplus k_0) \oplus (m_1 \oplus k_0) = m_0 \oplus m_1$$

- We can guess the value of k1 and compute the values of v0 and v1 directly from c0 and c1 .
- Since S[·] is publicly known and invertible, we can compute S^{-1}[v0] and S^{-1}[v1] .
- For the correct value of k1 , the cryptanalyst does know that

$$u_0 \oplus u_1 = S^{-1}[v_0] \oplus S^{-1}[v_1]$$

Cipher Two

- We can work backwards and guess the value of k2 to compute x0 and x1, and thus w0 and w1.
- We don’ t know k1, but we can compute $v0 \oplus v1$
- Starting from m0 and m1, we also know u0⊕u1

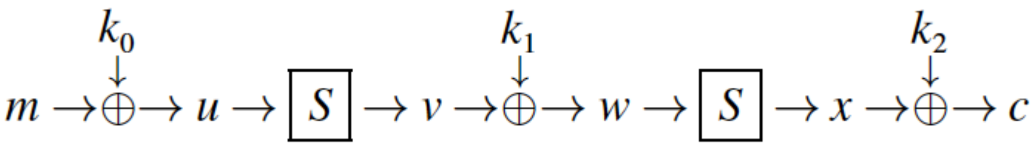
$$u0 \oplus u1 \rightarrow S \rightarrow \ v0 \oplus v1$$

Cannot be determined uniquely!!

$\text{CIPHERTWO}(m_0, k_0 \| k_1 \| k_2)$	$\text{CIPHERTWO}(m_1, k_0 \| k_1 \| k_2)$
$u_0 = m_0 \oplus k_0$	$u_1 = m_1 \oplus k_0$
$v_0 = S[u_0]$	$v_1 = S[u_1]$
$w_0 = v_0 \oplus k_1$	$w_1 = v_1 \oplus k_1$
$x_0 = S[w_0]$	$x_1 = S[w_1]$
$c_0 = x_0 \oplus k_2$	$c_1 = x_1 \oplus k_2$

$$c = S[S[m \oplus k_0] \oplus k_1] \oplus k_2$$

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S[x]$	6	4	c	5	0	7	2	e	1	f	3	d	8	a	9	b



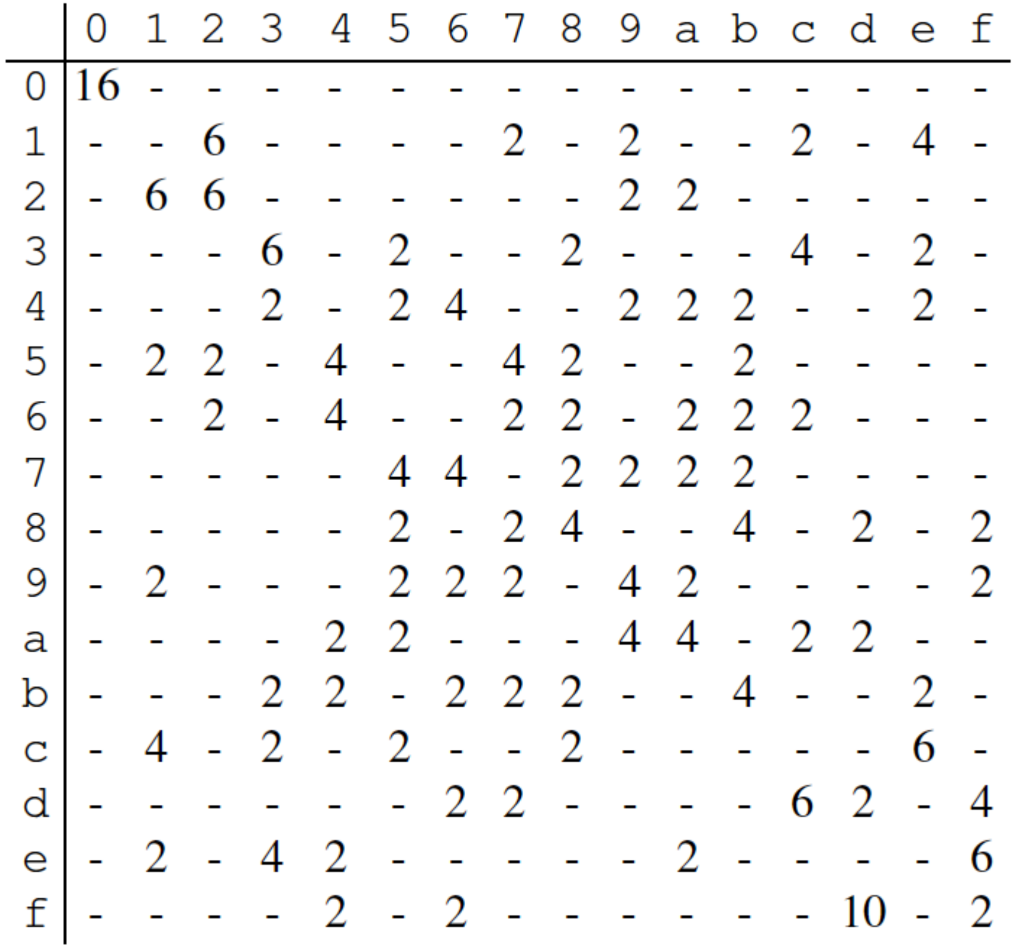
Inputs and output relations for i and $j = i \oplus \mathtt{f}$ across $S[\cdot]$.

i	j	$S[i]$	$S[j]$	$S[i] \oplus S[j]$
0	f	6	b	d
1	e	4	9	d
2	d	c	a	6
3	c	5	8	d
4	b	0	d	d
5	a	7	3	4
6	9	2	f	d
7	8	e	1	f
8	7	1	e	f
9	6	f	2	d
a	5	3	7	4
b	4	d	0	d
c	3	8	5	d
d	2	a	c	6
e	1	9	4	d
f	0	b	6	d

If $u0 \oplus u1 = f$, then Pr ($S[u_0] \oplus S[u_1] = \mathtt{d}$)= 10/16

Correct guess of k2 will let us find the match 10 times out of 16,
While incorrect guess will result in random behavior (1/16)

Differential Table



	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	-	6	-	-	-	-	2	-	2	-	-	2	-	4	-
2	-	6	6	-	-	-	-	-	-	2	2	-	-	-	-	-
3	-	-	-	6	-	2	-	-	2	-	-	-	4	-	2	-
4	-	-	-	2	-	2	4	-	-	2	2	2	-	-	2	-
5	-	2	2	-	4	-	-	4	2	-	-	2	-	-	-	-
6	-	-	2	-	4	-	-	2	2	-	2	2	2	-	-	-
7	-	-	-	-	-	4	4	-	2	2	2	2	-	-	-	-
8	-	-	-	-	-	2	-	2	4	-	-	4	-	2	-	2
9	-	2	-	-	-	2	2	2	-	4	2	-	-	-	-	2
a	-	-	-	-	2	2	-	-	-	4	4	-	2	2	-	-
b	-	-	-	2	2	-	2	2	2	-	-	4	-	-	2	-
c	-	4	-	2	-	2	-	-	2	-	-	-	-	-	6	-
d	-	-	-	-	-	-	2	2	-	-	-	-	6	2	-	4
e	-	2	-	4	2	-	-	-	-	-	2	-	-	-	-	6
f	-	-	-	-	2	-	2	-	-	-	-	-	-	10	-	2

The difference distribution table for S[·]. There is a row for each input difference d_{in} and the frequency with which a given output difference d_{out} occurs is given across the row. The entry (d_{in},d_{out}) divided by 16 gives the probability that a difference d_{in} gives difference d_{out} when taken over all possible pairs with difference d_{in}

Differential Characteristics

A pair (α,β) for which two inputs with difference α lead to two outputs with difference β is called a (differential) characteristic across the operation S[·]

$$\alpha \xrightarrow{S} \beta.$$

For example: Pr ($\mathtt{f} \xrightarrow{S} \mathtt{d}$) = 10/16

How about combining two S-Boxes?

$\mathtt{f} \xrightarrow{S} \mathtt{d}$

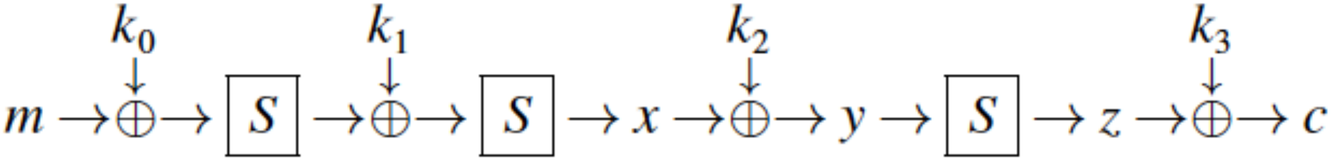
10/16

$\mathtt{d} \xrightarrow{S} \mathtt{c}$

6/16

⇓

Pr($\mathtt{f} \xrightarrow{S} \mathtt{d} \xrightarrow{S} \mathtt{c}$)= 10/16 x 6/16



Thus an attacker who chooses pairs of messages related by the difference f can expect the difference y0 ⊕y1 to take the value c with probability 15/64 > 4/64

Cipher Four

$$\left.\begin{array}{l}(\alpha_1,\alpha_2,\alpha_3,\alpha_4) \xrightarrow{S} (\beta_1,\beta_2,\beta_3,\beta_4) \\ \\ (\beta_1,\beta_2,\beta_3,\beta_4) \xrightarrow{P} (\gamma_1,\gamma_2,\gamma_3,\gamma_4)\end{array}\right\} (\alpha_1,\alpha_2,\alpha_3,\alpha_4) \xrightarrow{\mathscr{R}} (\gamma_1,\gamma_2,\gamma_3,\gamma_4)$$

First path

$$1 \left\{\begin{array}{l}(0,0,0,\mathtt{f}) \xrightarrow{S} (0,0,0,\mathtt{d}) \\ (0,0,0,\mathtt{d}) \xrightarrow{P} (1,1,0,1)\end{array}\right. \quad (0,0,0,\mathtt{f}) \xrightarrow{\mathscr{R}} (1,1,0,1)$$

$$\frac{10}{16} \times \left(\frac{6}{16}\right)^3 = \frac{135}{4096}$$

$$2 \left\{\begin{array}{l}(1,1,0,1) \xrightarrow{S} (2,2,0,2) \\ (2,2,0,2) \xrightarrow{P} (0,0,\mathtt{d},0)\end{array}\right. \quad (1,1,0,1) \xrightarrow{\mathscr{R}} (0,0,\mathtt{d},0).$$

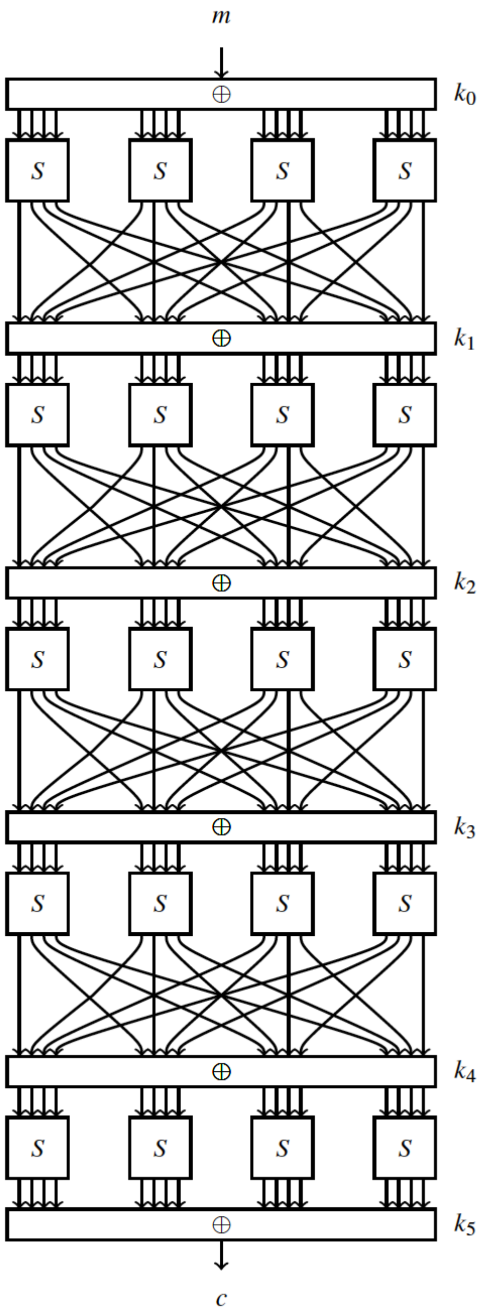
Not Good!!

Second path $(0,0,2,0) \xrightarrow{S} (0,0,2,0)$ and $(0,0,2,0) \xrightarrow{P} (0,0,2,0)$ $\quad (0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0)$

Two rounds: $(0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0)$ $\quad \left(\frac{6}{16}\right)^2.$

Four rounds: $(6/16)^4 = 0.02 < 1/16 = 0.06$ **Problem?**

Cipher Four



Differentials

- There could be more than one paths connecting $\alpha \to \beta$

$$(0,0,2,0) \xrightarrow{\mathscr{R}} ? \xrightarrow{\mathscr{R}} ? \xrightarrow{\mathscr{R}} ? \xrightarrow{\mathscr{R}} (0,0,2,0)$$

Four paths

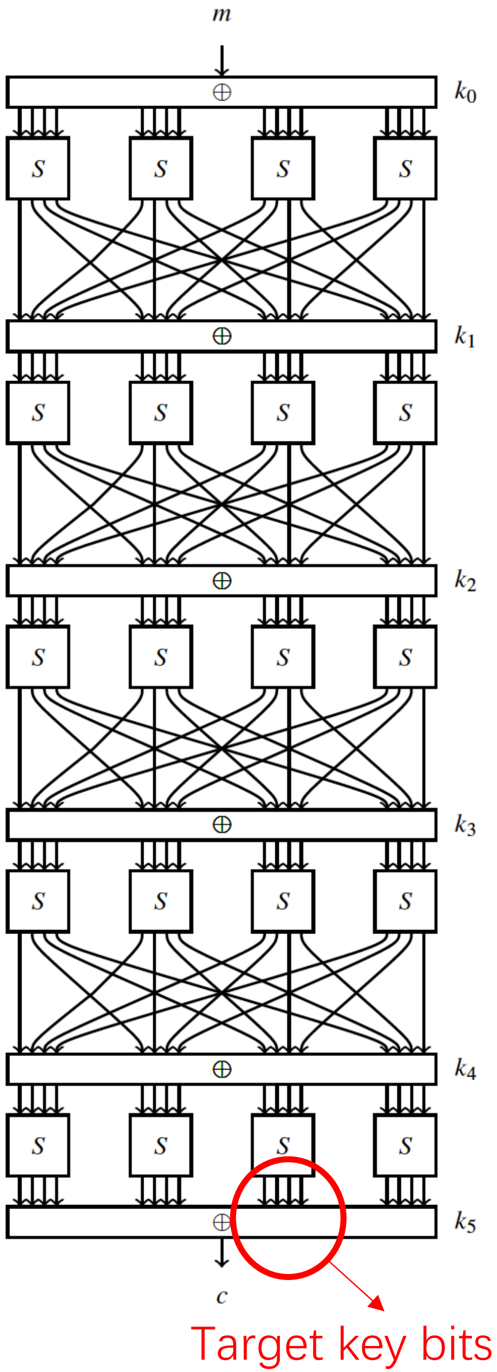
$$\left\{\begin{array}{l}
(0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0) \\
(0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,0,2) \xrightarrow{\mathscr{R}} (0,0,0,1) \xrightarrow{\mathscr{R}} (0,0,1,0) \xrightarrow{\mathscr{R}} (0,0,2,0), \\
(0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,0,2) \xrightarrow{\mathscr{R}} (0,0,1,0) \xrightarrow{\mathscr{R}} (0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0), \\
(0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,2,0) \xrightarrow{\mathscr{R}} (0,0,0,2) \xrightarrow{\mathscr{R}} (0,0,1,0) \xrightarrow{\mathscr{R}} (0,0,2,0).
\end{array}\right.$$

Now the probability becomes $4 \times (\frac{6}{16})^4 = \frac{81}{1024}.$

Recovering the key bits

- Use differential with prob 0.08 $(0,0,2,0) \xrightarrow{\mathscr{R}} ? \xrightarrow{\mathscr{R}} ? \xrightarrow{\mathscr{R}} ? \xrightarrow{\mathscr{R}} (0,0,2,0)$
- We assume that a pair survives the filtering process with probability 7387/65536 ≃ 0.11
- Attacker receives the encryption of t message pairs which satisfy the starting difference (0, 0, 2, 0) .
- t x 7387/65536 ≃ t x 0.11 pairs to survive filtering
- Pairs that satisfy the differential and there will be t x 0.08
- Over t chosen message pairs, we would expect roughly t x (0.11− 0.08) =t x 0.03 incorrect values for the target bits to be suggested.

If t=500, correct key bits will be suggested 500x0.08=40 times, while wrong key bits will be suggested 500x0.03=15 times. Thus we can recover the right one.



Linear Cryptanalysis – The idea

$$c = S[m \oplus k_0] \oplus k_1$$

Assume that an attacker knows a message m and the corresponding ciphertext c .

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S[x]$	f	e	b	c	6	d	7	8	0	3	9	a	4	2	1	5

$$u = m \oplus k_0,\ v = S[u], \text{ and } c = v \oplus k_1$$

$$\begin{tikzcd}
 & k_0 \arrow[d] & & & & k_1 \arrow[d] & \\
m \arrow[r] & \oplus \arrow[r] & u \arrow[r] & \boxed{S} \arrow[r] & v \arrow[r] & \oplus \arrow[r] & c
\end{tikzcd}$$

We view our blocks of input, output, and key as column vectors of bits. So if we wish to identify specific bits of vector x we can do so by pre-multiplying column vector by a row vector which acts as a mask

$$(1,0,0,0) \times \begin{pmatrix} m_3 \\ m_2 \\ m_1 \\ m_0 \end{pmatrix} = m_3, \text{ and } (0,0,1,0) \times \begin{pmatrix} m_3 \\ m_2 \\ m_1 \\ m_0 \end{pmatrix} = m_1$$

$$(1,0,1,1) \times \begin{pmatrix} m_3 \\ m_2 \\ m_1 \\ m_0 \end{pmatrix} \oplus (1,0,1,1) \times \begin{pmatrix} k_3 \\ k_2 \\ k_1 \\ k_0 \end{pmatrix} = m_3 \oplus m_1 \oplus m_0 \oplus k_3 \oplus k_1 \oplus k_0$$

Linear Cryptanalysis – The idea

$$\begin{tikzcd}
 & k_0 \arrow[d] & & & & k_1 \arrow[d] & \\
m \arrow[r] & \oplus \arrow[r] & u \arrow[r] & \boxed{S} \arrow[r] & v \arrow[r] & \oplus \arrow[r] & c
\end{tikzcd}$$

$c_3 = m_3 \oplus m_1 \oplus m_0 \oplus k_3 \oplus k_1 \oplus k_0,$ Can be written by using mask

$\alpha \cdot c = \beta \cdot m \oplus \beta \cdot k,$ $\alpha = (1,0,0,0)$ and $\beta = (1,0,1,1).$

Pr ($\alpha \cdot c = \beta \cdot m \oplus \beta \cdot k,$)≠ 1/2

$$\begin{aligned}
(\alpha \cdot m) &= (\alpha \cdot k_0) \oplus (\alpha \cdot u) && \text{with probability } 1 \\
(\alpha \cdot u) &= (\beta \cdot v) && \text{with probability } p \\
(\beta \cdot v) &= (\beta \cdot k_1) \oplus (\beta \cdot c) && \text{with probability } 1.
\end{aligned}$$

We can just add these equations together to get

$$(\alpha \cdot m) \oplus (\alpha \cdot u) \oplus (\beta \cdot v) = (\alpha \cdot k_0) \oplus (\alpha \cdot u) \oplus (\beta \cdot v) \oplus (\beta \cdot k_1) \oplus (\beta \cdot c),$$

$$\Downarrow$$

$$\underline{(\alpha \cdot m) \oplus (\beta \cdot c)} = \underline{(\alpha \cdot k_0) \oplus (\beta \cdot k_1)} \quad \text{with probability } p$$

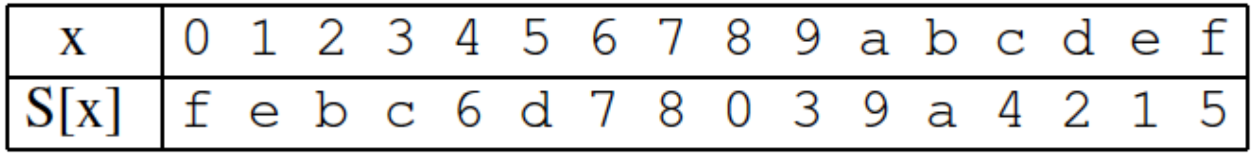
Message and ciphertext

P=0
P=1

We are happy at both cases. Why?

Non-linear part

S-Box

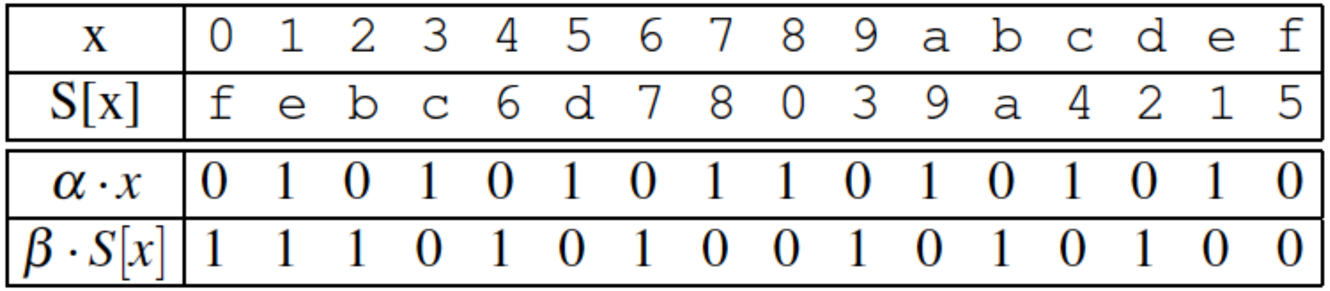


x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S[x]	f	e	b	c	6	d	7	8	0	3	9	a	4	2	1	5

mask

$$\alpha = (1,0,0,1) \text{ and } \beta = (0,0,1,0)$$

Count the number of times that $\alpha \cdot x = \beta \cdot S[x]$



x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S[x]	f	e	b	c	6	d	7	8	0	3	9	a	4	2	1	5
$\alpha \cdot x$	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
$\beta \cdot S[x]$	1	1	1	0	1	0	1	0	0	1	0	1	0	1	0	0

What is the probability? 14/16

$$(\alpha \cdot m) \oplus (\beta \cdot c) \oplus 1 = (\alpha \cdot k_0) \oplus (\beta \cdot k_1)$$

- We use two counters T0 and T1 which are initialized to T0 = T1 = 0
- Increment the counter T0 by 1 if evaluate the left-hand side of the equation to 0
- Increment the counter T1 by 1 if evaluate the left-hand side of the equation to 1
- Request the encryptions of N known plaintexts
- Count the number of 1s on the left side of the equation.
- If (α ·k0)⊕(β ·k1) = 1 , then our counter T0 should have the value 2N/16, and T1 should be 14N/16
- Determine one bit of the key

Joining Approximation

$$\begin{tikzcd}
 & k_0 \arrow[d] & & & & k_1 \arrow[d] & & & & k_2 \arrow[d] & \\
m \arrow[r] & \oplus \arrow[r] & u \arrow[r] & \boxed{S} \arrow[r] & v \arrow[r] & \oplus \arrow[r] & w \arrow[r] & \boxed{S} \arrow[r] & x \arrow[r] & \oplus \arrow[r] & c
\end{tikzcd}$$

$$\begin{aligned}
(\alpha \cdot m) &= (\alpha \cdot k_0) \oplus (\alpha \cdot u),\\
(\beta \cdot v) &= (\beta \cdot k_1) \oplus (\beta \cdot w),\\
(\gamma \cdot x) &= (\gamma \cdot k_2) \oplus (\gamma \cdot c).
\end{aligned}$$

$$\begin{aligned}
\alpha \cdot u = \beta \cdot S[u] = \beta \cdot v \quad &\text{with probability } p_1 \neq \tfrac{1}{2} \text{ and}\\
\beta \cdot w = \gamma \cdot S[w] = \gamma \cdot x \quad &\text{with probability } p_2 \neq \tfrac{1}{2}.
\end{aligned}$$

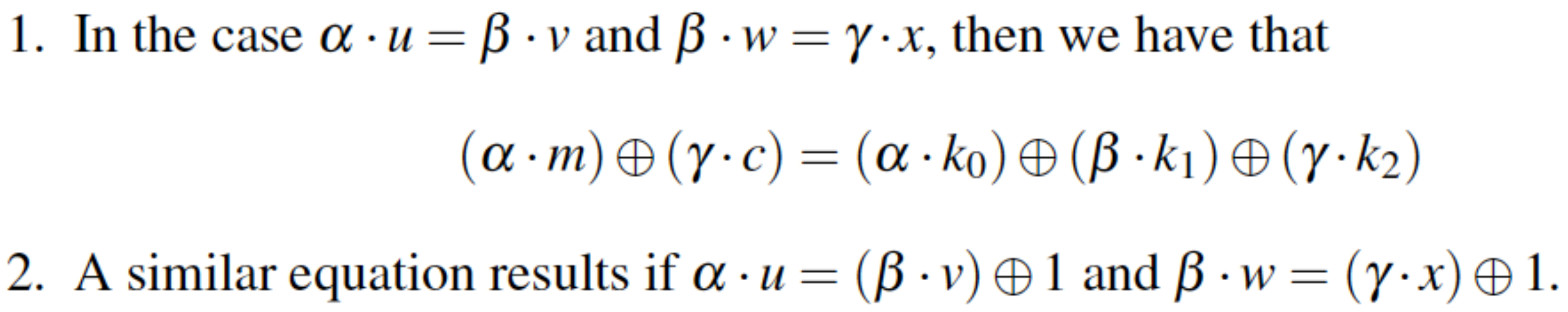
$$\Downarrow$$

$$(\alpha \cdot m) \oplus (\gamma \cdot c) \oplus (\alpha \cdot u) \oplus (\beta \cdot v) \oplus (\beta \cdot w) \oplus (\gamma \cdot x) = (\alpha \cdot k_0) \oplus (\beta \cdot k_1) \oplus (\gamma \cdot k_2),$$

$$\begin{aligned}
(\alpha \cdot u) &= (\beta \cdot v) \text{ with probability } p_1\\
(\beta \cdot w) &= (\gamma \cdot x) \text{ with probability } p_2,
\end{aligned}$$

What is the total probability that the following equation hold?

$$(\alpha \cdot m) \oplus (\gamma \cdot c) = (\alpha \cdot k_0) \oplus (\beta \cdot k_1) \oplus (\gamma \cdot k_2)$$



1. In the case $\alpha \cdot u = \beta \cdot v$ and $\beta \cdot w = \gamma \cdot x$, then we have that

$$(\alpha \cdot m) \oplus (\gamma \cdot c) = (\alpha \cdot k_0) \oplus (\beta \cdot k_1) \oplus (\gamma \cdot k_2)$$

2. A similar equation results if $\alpha \cdot u = (\beta \cdot v) \oplus 1$ and $\beta \cdot w = (\gamma \cdot x) \oplus 1$.

$$p_1 \times p_2$$

$$(1 - p_1) \times (1 - p_2).$$

Together: $$p_1 p_2 + (1 - p_1)(1 - p_2).$$

Piling-up lemma and Linear Approximation Table

Matsui

- Know $\Pr(V_i = 0) = ½ + e_i$
- $\Pr(V_1 \oplus V_2 \oplus \cdots \oplus V_n = 0) = ½ + 2^{n-1}\prod e_i$
- V_i' s are independent random variables
- e_i is the bias $-½ \leq e_i \leq ½$

Use to combine linear equations if view each as independent random variable

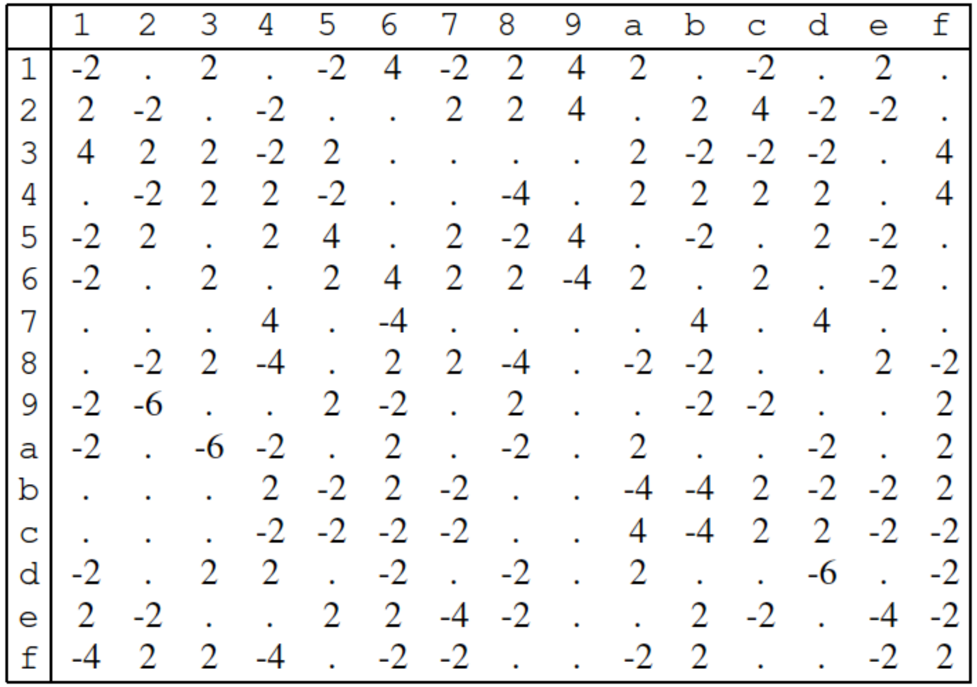
By choosing $\alpha = \beta = \gamma = \mathbf{d}$,

$$(\alpha \cdot m) \oplus (\gamma \cdot c) = (\alpha \cdot k_0) \oplus (\beta \cdot k_1) \oplus (\gamma \cdot k_2)$$

holds with probability $\frac{1}{8} \times \frac{1}{8} + \frac{7}{8} \times \frac{7}{8} = \frac{25}{32} = \frac{1}{2} + \frac{9}{32}$.

$N = |p - 1/2|^{-2}$ Messages are required!

Linear Approximation Table



	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	-2	.	2	.	-2	4	-2	2	4	2	.	-2	.	2	.
2	2	-2	.	-2	.	.	2	2	4	.	2	4	-2	-2	.
3	4	2	2	-2	2	.	.	.	.	2	-2	-2	-2	.	4
4	.	-2	2	2	-2	.	.	-4	.	2	2	2	2	.	4
5	-2	2	.	2	4	.	2	-2	4	.	-2	.	2	-2	.
6	-2	.	2	.	2	4	2	2	-4	2	.	2	.	-2	.
7	.	.	.	4	.	-4	.	.	.	.	4	.	4	.	.
8	.	-2	2	-4	.	2	2	-4	.	-2	-2	.	.	2	-2
9	-2	-6	.	.	2	-2	.	2	.	.	-2	-2	.	.	2
a	-2	.	-6	-2	.	2	.	-2	.	2	.	.	-2	.	2
b	.	.	.	2	-2	2	-2	.	.	-4	-4	2	-2	-2	2
c	.	.	.	-2	-2	-2	-2	.	.	4	-4	2	2	-2	-2
d	-2	.	2	2	.	-2	.	-2	.	2	.	.	-6	.	-2
e	2	-2	.	.	2	2	-4	-2	.	.	2	-2	.	-4	-2
f	-4	2	2	-4	.	-2	-2	.	.	-2	2	.	.	-2	2

If we divide entry (i, j) by 16 and add 1/2 then this gives the probability that an input masked by i equals the output masked by j

Cipher D

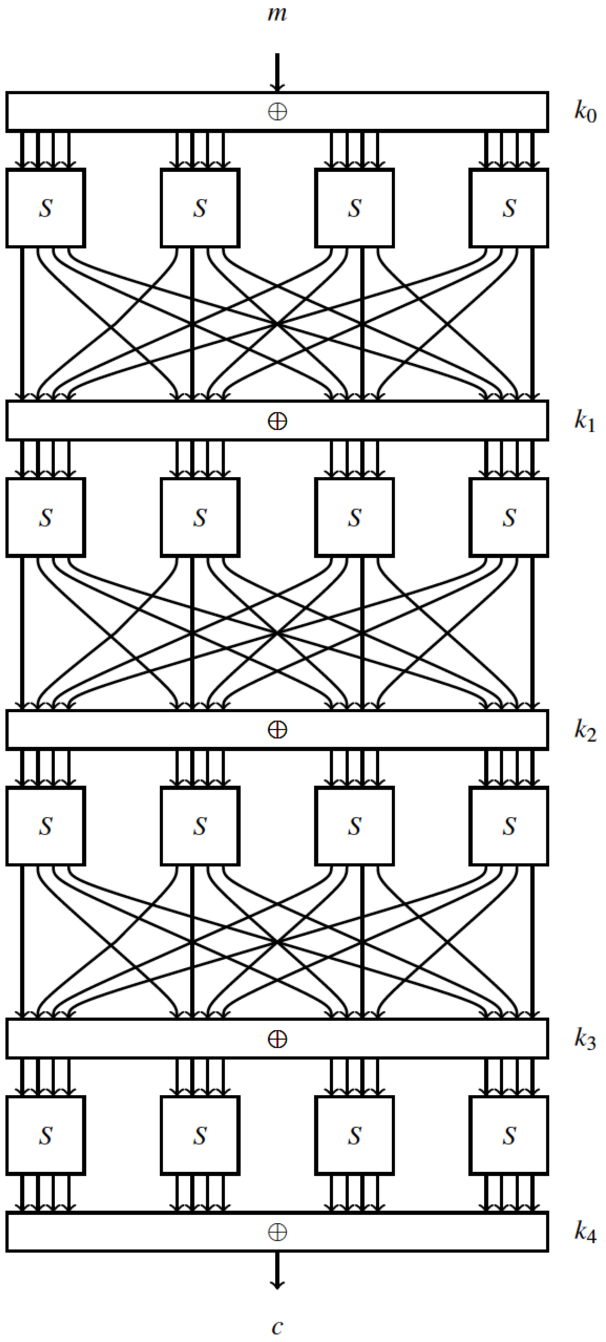
First Round $(000\mathtt{d}) \xrightarrow{S} (000\mathtt{d})$ $\frac{1}{2} - \frac{6}{16}$

$(000\mathtt{d}) \xrightarrow{\mathscr{R}} (1101)$

Second Round $(1101) \xrightarrow{S} (6606) \xrightarrow{P} (0\mathtt{d}\mathtt{d}0)$, $\frac{1}{2} + 2^2\left(\frac{4}{16}\right)^3 = \frac{1}{2} + \frac{1}{16}$

$(000\mathtt{d}) \xrightarrow{\mathscr{R}} (1101) \xrightarrow{\mathscr{R}} (0\mathtt{d}\mathtt{d}0)$ ½+2x(-6/16 x 1/16) = ½ + 3/64

We can continue to go on, but result is not good



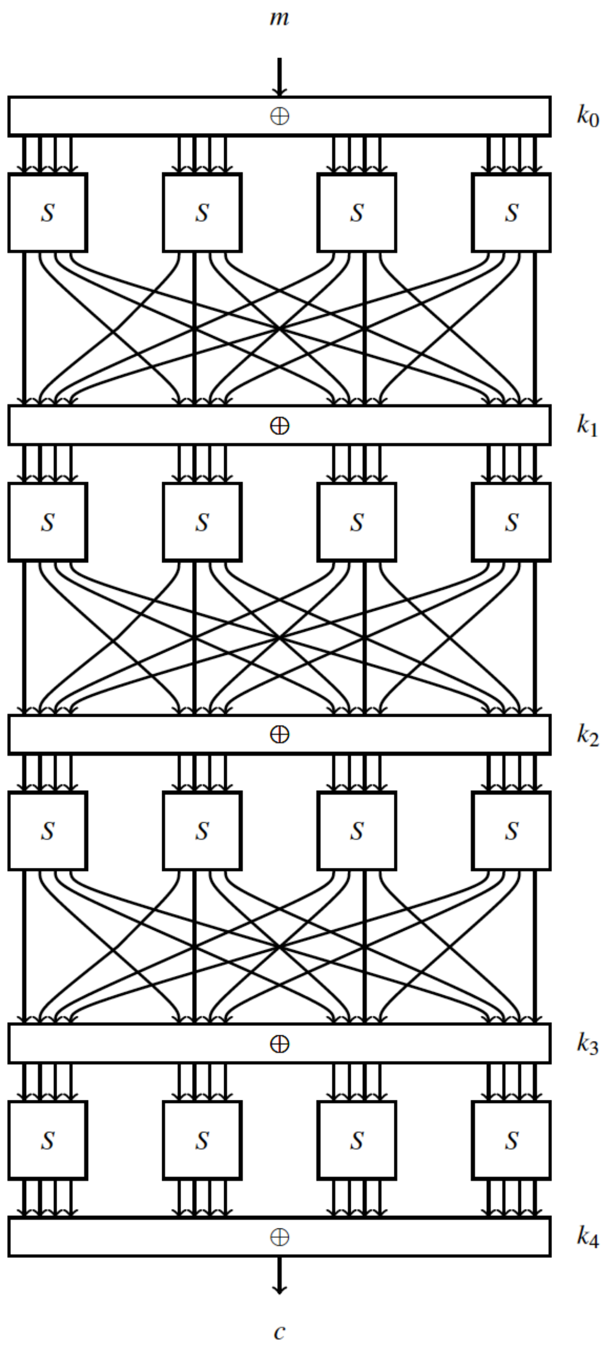
Cipher D

$$(8000) \xrightarrow{\mathscr{R}} (8000) \qquad \frac{1}{2} - \frac{4}{16}.$$

$$(8000) \xrightarrow{\mathscr{R}} (8000) \xrightarrow{\mathscr{R}} (8000) \qquad \left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2 = \frac{5}{8} = \frac{1}{2} + \frac{1}{8}.$$

$$(8000) \xrightarrow{\mathscr{R}} (8000) \xrightarrow{\mathscr{R}} (8000) \xrightarrow{\mathscr{R}} (8000) \xrightarrow{\mathscr{R}} (8000)$$

$$\frac{1}{2} + 2^3 \left(\frac{1}{4}\right)^4 = \frac{17}{32} = \frac{1}{2} + \frac{1}{32}$$



Linear Hull

Again, more than one paths

$$(8,0,0,0) \xrightarrow{\mathscr{R}} (*,*,*,*) \xrightarrow{\mathscr{R}} (*,*,*,*) \xrightarrow{\mathscr{R}} (*,*,*,*) \xrightarrow{\mathscr{R}} (8,0,0,0)$$

For example:

$$(8000) \xrightarrow{\mathscr{R}} (0800) \xrightarrow{\mathscr{R}} (4000) \xrightarrow{\mathscr{R}} (8000) \xrightarrow{\mathscr{R}} (8000)$$

$$(8000) \xrightarrow{\mathscr{R}} (8000) \xrightarrow{\mathscr{R}} (0800) \xrightarrow{\mathscr{R}} (4000) \xrightarrow{\mathscr{R}} (8000)$$

Key Recovery and Data Complexity

Assume the approximation: $(m \cdot \alpha) \oplus (c \cdot \beta) = (k \cdot \gamma)$ and $p = \frac{1}{2} + \varepsilon$ $\varepsilon > 0$.

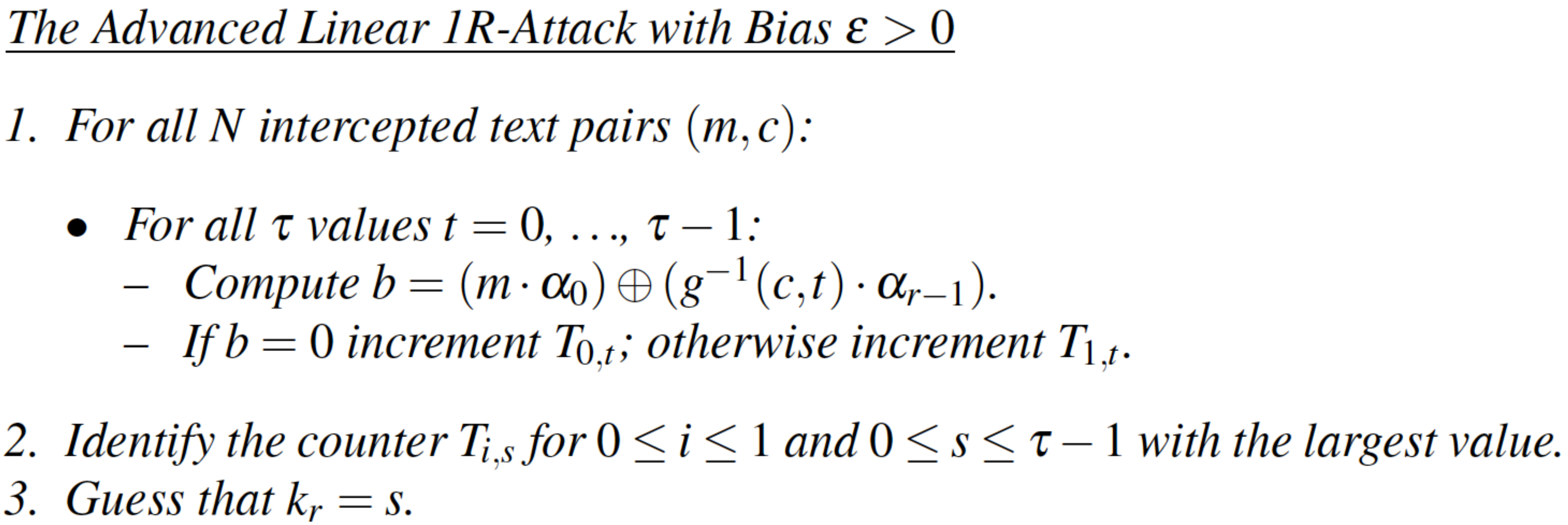
Given N plaintexts m and corresponding ciphertexts c we can recover one key bit as follows. Let T_0 denote the number of times $(m \cdot \alpha) \oplus (c \cdot \beta)$ is equal to 0 while T_1 denotes the number of times $(m \cdot \alpha) \oplus (c \cdot \beta)$ is equal to 1.

The Basic Linear Attack with Characteristic of Bias $\varepsilon > 0$

1. *For all N intercepted texts (m, c):*
   - *Compute $b = (m \cdot \alpha) \oplus (c \cdot \beta)$.*
     – *If $b = 0$ increment counter T_0. Otherwise increment counter T_1.*
2. *If $T_0 > \frac{N}{2}$ guess that $k \cdot \gamma = 0$. Otherwise guess that $k \cdot \gamma = 1$.*

N plaintexts	$\frac{\varepsilon^{-2}}{16}$	$\frac{\varepsilon^{-2}}{8}$	$\frac{\varepsilon^{-2}}{4}$	$\frac{\varepsilon^{-2}}{2}$	ε^{-2}
success rate	69%	76%	84%	92%	98%

Last round attack

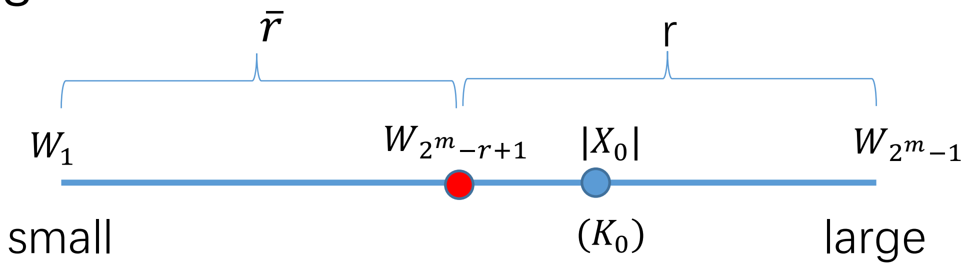


The Advanced Linear 1R-Attack with Bias $\varepsilon > 0$

```
1. For all N intercepted text pairs $(m, c)$:
    • For all $\tau$ values $t = 0, \ldots, \tau - 1$:
        – Compute $b = (m \cdot \alpha_0) \oplus (g^{-1}(c, t) \cdot \alpha_{r-1})$.
        – If $b = 0$ increment $T_{0,t}$; otherwise increment $T_{1,t}$.
2. Identify the counter $T_{i,s}$ for $0 \leq i \leq 1$ and $0 \leq s \leq \tau - 1$ with the largest value.
3. Guess that $k_r = s$.
```

Successful probability of Linear Cryptanalysis

- Right key key ranks the top r among 2^m keys
- m-bit key is attacked
- Approximation probability is p
- Using n data blocks
- k_0 is the right key, $k_i, 1 \leq i \leq 2^m - 1$
- T_i is the counter for the plaintexts satisfying the approximation with key k_i
- $X_i = \frac{T_i}{N} - \frac{1}{2}$, $Y_i = |X_i|$
- W_i be Y_i sorted in increasing order



Advantage: a=m-lgr

Successful attack: $\frac{X_0}{p-\frac{1}{2}} > 0$ and $|X_0| > W_{n-r+1}$

Distribution of some random variables

For the right key $\boldsymbol{K_0}$:

Assume $T_0 = \sum C_i$ where $C_i \sim Bernouli(p)$, so we have $T_0 \sim B(n, p) \approx N(np, np(1-p))$

$$X_0 \sim N(p - \frac{1}{2}, p(1-p)/n) \approx N(p - \frac{1}{2}, 1/4n)$$

For the wrong keys $\boldsymbol{K_i}$:

Assume zero bias for the wrong keys where $p = 1/2$,
$Y_i, i \neq 0 \sim FN(\mu_w, \sigma_w^2) = FN(0, 1/4n)$

FN: folded normal distribution

Theorem (Order statistic). Let $\bar{\boldsymbol{r}} = 2^m - 2^a$, $W_{\bar{r}} \sim N(\mu_q, \sigma_q^2)$

Random variable	Cumulative function	Density function
Y_i	F_w	f_w
X_0	F_0	f_0
$W_{\bar{r}}$	F_q	f_q

$$\mu_q = F_w^{-1}\left(1 - 2^{-a}\right) = \mu_w + \sigma_w \Phi^{-1}\left(1 - 2^{-a-1}\right)$$

$$\sigma_q = \frac{1}{f_w\left(\mu_q\right)} 2^{-\frac{m+a}{2}} = \frac{\sigma_w}{2\phi\left(\Phi^{-1}\left(1 - 2^{-a-1}\right)\right)} 2^{-\frac{m+a}{2}}$$

Probability derivation

- Assume p > ½, then an a-bit advantage attack on an m-bit key is defined as

$$X_0 > 0 \qquad and \qquad X_0 > W_{\bar{r}}$$

The success probability Ps is $$P_S = \int_0^\infty \int_{-\infty}^x f_q(y)dy f_0(x)dx$$

Since $W_{\bar{r}} < 0$ is negligible, the successful conditions can be simplified as $X_0 > W_{\bar{r}}$

$$X_0 - W_{\bar{r}} \sim N(\mu_0 - \mu_q, \sigma_0^2 + \sigma_q^2)$$

$$\begin{aligned} P_S &= P(X_0 - W_{\bar{r}} \ > \ 0) \\ &= \int_0^\infty f_J(x)\,dx \\ &= \int_{-\frac{\mu_0 - \mu_q}{\sqrt{\sigma_0^2 + \sigma_q^2}}}^\infty \phi(x)\,dx \end{aligned}$$

$\Rightarrow$

Assume

$$\sqrt{\sigma_0^2 + \sigma_q^2} \approx \sigma_0.$$

$\Rightarrow$

$$\begin{aligned} P_S &= \int_{-\frac{\mu_0 - \mu_q}{\sigma_0}}^\infty \phi(x)\,dx \\ &= \int_{-2\sqrt{N}(|p-1/2|-F_w^{-1}(1-2^{-a}))}^\infty \phi(x)\,dx \end{aligned}$$

Successful Probability

Theorem. Let Ps be the probability that a linear attack on an m-bit subkey, with a linear approximation of probability p, with n known plaintext blocks, delivers an a-bit or higher advantage. Assuming that the linear approximation’ s probability to hold is independent for each key tried and is equal to 1/2 for all wrong keys, we have, for sufficiently large m and n,

$$P_S = \Phi\left(2\sqrt{N}|p - 1/2| - \Phi^{-1}(1 - 2^{-a-1})\right)$$

⬇

$$N = \left(\frac{\Phi^{-1}(P_S) + \Phi^{-1}(1 - 2^{-a-1})}{2}\right)^2 \cdot |p - 1/2|^{-2}$$

Other Cryptanalysis methods

- Multi-differential attack
- Multi-Linear attack
- Boomerang attack
- Impossible differential attack
- Truncated differential attack
- Meet-in-the-Middle attack

Statistical Test

- Sixteen tests performed on eight sets of data for each cipher.
  - Do not prove cipher is secure
  - Failing a test indicates a weakness
  - NIST AES competition finalists: > 96.33% of cases passing
- What if cipher fails a test?
  - Some relationship between P,C,K – but don’ t know exactly what
  - Example, key with a 1 in bit j may be prone to produce ciphertext with more 0’ s than 1’ s.

Statistical Test

- Frequency (Monobit): are proportions of 0's and 1's in the bit sequence close enough to ½ .
- Frequency within a Block: Frequency test applied to fixed-sized blocks within the bit sequence.
- Runs: The number of runs (sequence of all 0's or all 1's) in the bit sequence is determined.
- Longest Run of Ones within a Block: The longest run of 1's within a block is determined.
- Binary Matrix Rank: 32-by-32 matrices are created from the bit sequence and their ranks computed. Determines if any linear dependence among fixed-length segments of bits within the sequence.
- Discrete Fourier Transform: determines if there are repetitive patterns in the bit sequence.
- Non-overlapping Template Matching: counts the number of times a m-bit pattern occurs in the bit sequence using a sliding window. The window slides 1 bit when no match and slides m bits when a match occurs so a bit will be involved in at most one match for a given pattern. Ex. m = 9
- Overlapping Template Matching: same as the previous test except that the window always slides 1 bit.

Statistical Test

- Maurer's Universal Statistical: determines if the bit sequence can be compressed based on the number of bits between occurrences of a pattern.
- Lempel-Ziv Compression: determines how much a bit sequence can be compressed based on the number of distinct patterns.
- Linear Complexity: Berlekamp-Massey algorithm is applied to a 1000 bit sequence to determine a linear feedback shift register that produces the sequence. The length of the LFRS indicates if the sequence is sufficiently random.
- Serial: The number of times each 2^m bit pattern occurs is determined, for some integer m.
- Approximate Entropy: The number of times each 2^m and each 2^(m+1) bit pattern is determined, for some integer m.
- Cumulative Sums: cumulative sum of the bits is computed for each position in the sequence. The sum is computed by adding -1 for each bit that is 0 and adding 1 for each bit that is 1.
- Random Excursions: number of times the cumulative sum crosses zero is determined.
- Random Excursions Variant: number of times the cumulative sum is a particular value is determined.