A Tutorial on Linear and Differential Cryptanalysis

Yichen Zhu

2021/05/24

A Basic Substitution-Permutation Network Cipher

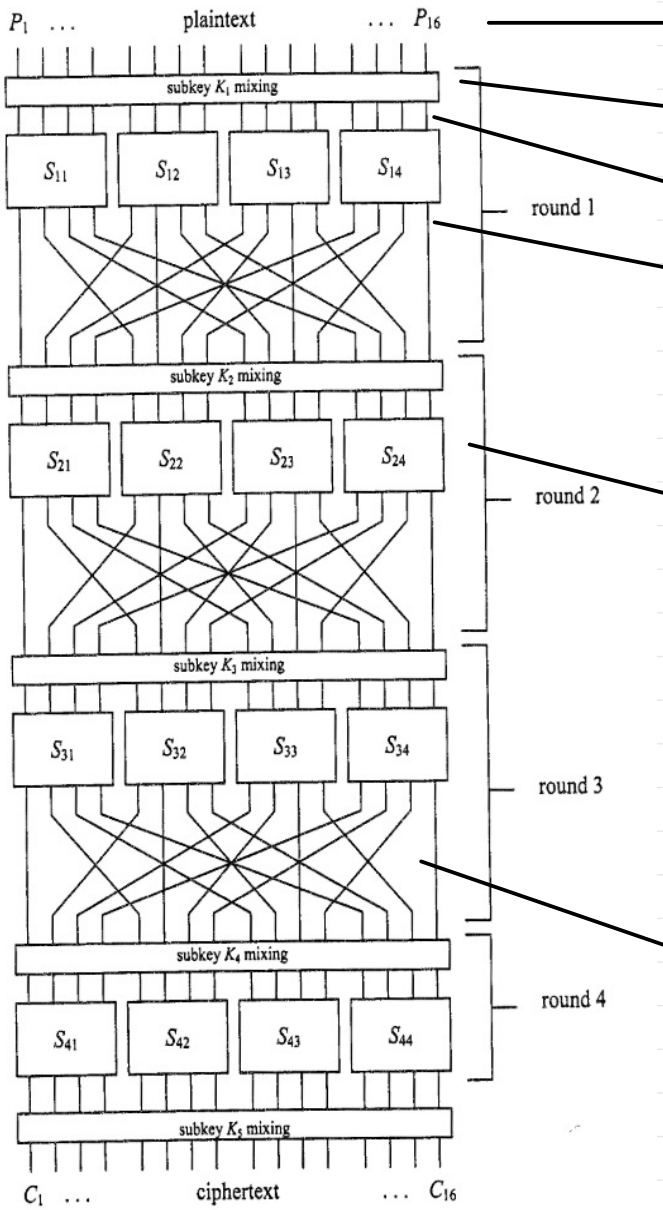

Figure 1. Basic Substitution-Permutation Network (SPN) Cipher.

block size : 16 bits

$U_i = P \oplus K_i$ → subkeys

input of S-box : U_i

output of S-box : V_i

1. generally, subkeys are generated from master key (key schedule)
2. in this case, subkeys are independently generated.

4×4 S-boxes → nonlinear mapping

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

Table 1. S-box Representation (in hexadecimal).

1. generally, S-boxes are different
2. in this case, all S-boxes are the same

Permutation

Piling-Up Principle

random binary variable X_1, X_2 $\quad Pr(X_i = 0) = p_i$

$$\begin{aligned} Pr(X_1 \oplus X_2 = 0) &= Pr(X_1 = X_2) \\ &= Pr(X_1 = 0, X_2 = 0) + Pr(X_1 = 1, X_2 = 1) \\ &= p_1 p_2 + (1-p_1)(1-p_2) \qquad \text{assume } X_1, X_2 \text{ are independent} \\ &= \frac{1}{2} + 2\varepsilon_1\varepsilon_2 \qquad \text{let } \varepsilon_i = p_i - \frac{1}{2} \end{aligned}$$

↘ the linear probability bias of $X_1 = 0$

(Piling-Up Lemma) n independent, random binary variables $X_1, X_2, \cdots, X_n$

$$Pr(X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0) = \frac{1}{2} + \underbrace{2^{n-1} \prod_{i=1}^{n} \varepsilon_i}_{\varepsilon_{1,2,\cdots,n}}$$

Corollary

$$(\forall i \in \{1,2,\cdots,n\} : p_i = 0) \rightarrow Pr(X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0) = 0$$

$$(\forall i \in \{1,2,\cdots,n\} : p_i = 1) \rightarrow Pr(X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0) = 1$$

$$(\exists i \in \{1,2,\cdots,n\} : p_i = \tfrac{1}{2}) \rightarrow Pr(X_1 \oplus X_2 \oplus \cdots \oplus X_n = 0) = \frac{1}{2}$$

3 independent, random binary variable X_1, X_2, X_3

$$\left.\begin{aligned} Pr(X_1 \oplus X_2 = 0) &= \tfrac{1}{2} + \varepsilon_{1,2} \\ Pr(X_2 \oplus X_3 = 0) &= \tfrac{1}{2} + \varepsilon_{2,3} \end{aligned}\right\} \Rightarrow Pr((X_1 \oplus X_2) \oplus (X_2 \oplus X_3) = 0) = \frac{1}{2} + 2\varepsilon_{1,2}\varepsilon_{2,3}$$

strictly, $X_1 \oplus X_2$ and $X_2 \oplus X_3$ are independent

$\Leftrightarrow \varepsilon_1 = 0$ or $\varepsilon_3 = 0$ or $\varepsilon_2 = \pm\frac{1}{2}$

treat as if $X_1 \oplus X_2$ and $X_2 \oplus X_3$ are independent

Linear Cryptanalysis (Known-Plaintext Attack)

Basic Idea

consider $Pr(X_1 \oplus X_2 \oplus \cdots \oplus X_n \oplus Y_1 \oplus Y_2 \oplus \cdots \oplus Y_n = 0) = ?$

- X, Y true random: $? = \frac{1}{2}$
- poor randomization: $\varepsilon = ? - \frac{1}{2}$ → linear probability bias

1. how to construct ?
2. how it can be exploited ?

consider the nonlinear component : the S-box

develop linear approximations

Analyzing the Cipher Components — linear vulnerabilities of an S-box

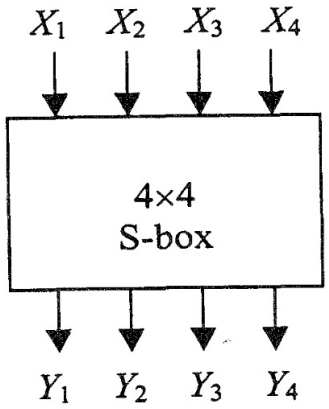

Figure 2. S-box Mapping.

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	$X_3 \oplus X_4$	$Y_1 \oplus Y_4$
				S'1	S'2	S'3	S'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

Table 3. Sample Linear Approximations of S-box.

e.g.

$Pr(X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4) = \frac{12}{16}$

$0 \cdot X_1 \oplus 1 \cdot X_2 \oplus 1 \cdot X_3 \oplus 0 \cdot X_4$

$(0110)_2 = (6)_{16}$

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

Table 4. Linear Approximation Table.

$f : \{0,1\}^n \to \{0,1\}$

$\varepsilon(f) = 2^n - 2 \cdot hw(f)$

$= 2^n - 2 \cdot \#\{x \mid f(x)=1\}$

$LAT(a,b)$: the bias of $a=b$

$$LAT(6,B) = \frac{\varepsilon(X_2 \oplus X_3 \oplus S_1'(X) \oplus S_3'(X) \oplus S_4'(X))}{2}$$

$= 2^3 - 4 = +4$

property :

1. $LAT(0,0) = +8$
2. $LAT(0,j) = 0 \quad (j \neq 0)$
3. $\forall i : \sum_j LAT(i,j) = +8 \text{ or } -8$

   $\forall j : \sum_i LAT(i,j) = +8 \text{ or } -8$

Constructing Linear Approximations for the Complete Cipher

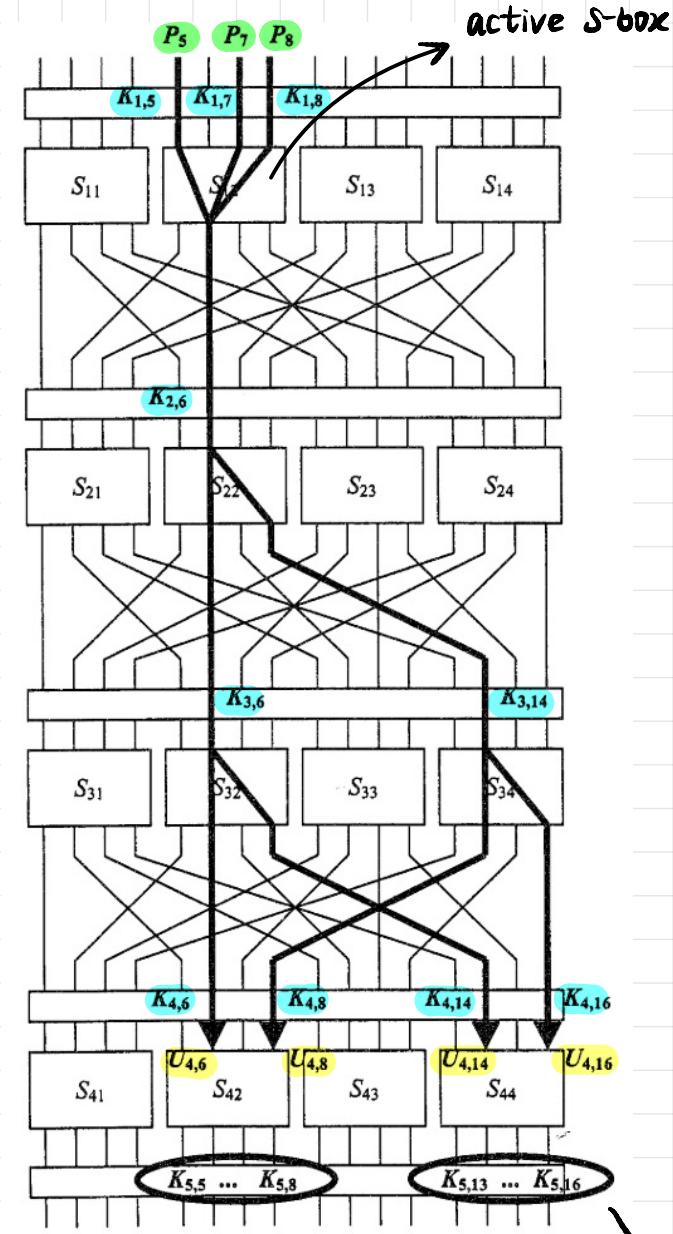

Figure 3. Sample Linear Approximation.

$$U_{1,5} \oplus U_{1,7} \oplus U_{1,8} = (P_5 \oplus K_{1,5}) \oplus (P_7 \oplus K_{1,7}) \oplus (P_8 \oplus K_{1,8})$$

$V_{1,6} = U_{1,5} \oplus U_{1,7} \oplus U_{1,8}$ (1) $\varepsilon_{12} = \frac{+4}{16}$

$V_{2,6} \oplus V_{2,8} = U_{2,6} = V_{1,6} \oplus K_{2,6}$ (2) $\varepsilon_{22} = \frac{-4}{16}$

$V_{3,6} \oplus V_{3,8} = U_{3,6} = V_{2,6} \oplus K_{3,6}$ (3) $\varepsilon_{32} = \frac{-4}{16}$

$V_{3,14} \oplus V_{3,16} = U_{3,14} = V_{2,8} \oplus K_{3,14}$ (4) $\varepsilon_{34} = \frac{-4}{16}$

$V_{3,6} = U_{4,6} \oplus K_{4,6}$, ...

assume S-boxes are independent

$$\bigoplus_i U_{4,i} \oplus \bigoplus_j P_j \oplus \bigoplus_k K_k = 0 \qquad \varepsilon = 2^3 \cdot \varepsilon_{12}\varepsilon_{32}\varepsilon_{32}\varepsilon_{34} = -\frac{1}{32}$$

k: 0 or 1

$$\bigoplus_i U_{4,i} \oplus \bigoplus_j P_j = 0$$

holds with probability $\frac{15}{32}$ or $\frac{17}{32}$

Extracting Key Bits

N_L (P, C)

$V_4 \oplus K_5 \rightarrow U_4 = S_{4,2}^{-1}(C \oplus K_5)$

K_5	count
0000 0000 0000 0000	0
0000 0000 0000 0001	0
...	...
0000 1111 0000 1111	0

if $\bigoplus_i U_{4,i} \oplus \bigoplus_j P_j = 0$, then ++count[k]

General Algorithm

1. determine a good linear approximation of $R-1$ rounds (How?)
2. collect N_L plaintext/ciphertext samples
3. For all k : count[k] = 0
4. For all k :
   For all (P, C) :
   if linear approximation holds :
   ++ count[k]

→ the correct partial subkey will result probability significantly different from $\frac{1}{2}$.

partial subkey $[K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}]$	|bias|	partial subkey $[K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}]$	|bias|
1 C	0.0031	2 A	0.0044
1 D	0.0078	2 B	0.0186
1 E	0.0071	2 C	0.0094
1 F	0.0170	2 D	0.0053
2 0	0.0025	2 E	0.0062
2 1	0.0220	2 F	0.0133
2 2	0.0211	3 0	0.0027
2 3	0.0064	3 1	0.0050
2 4	**0.0336**	3 2	0.0075
2 5	0.0106	3 3	0.0162
2 6	0.0096	3 4	0.0218
2 7	0.0074	3 5	0.0052
2 8	0.0224	3 6	0.0056
2 9	0.0054	3 7	0.0048

Table 5. Experimental Results for Linear Attack.

$|bias| = \frac{|count - \frac{N_L}{2}|}{N_L}$

$0.0336 \approx \frac{1}{32} = 0.03125$

Complexity of Attack.

the number of active S-boxes ↑
the bias of active S-boxes ↓
⇒ the bias of linear approximation ↓

use data required when considering the complexity of the cryptanalysis.

in practice. $N_L \approx \varepsilon^{-2}$

⚠ two assumptions

1. each S-box approximation is independent
2. one linear approximation scenario is sufficient to determine the best linear expression

↘ linear hull.

Differential Cryptanalysis (Chosen-Plaintext Attack)

Basic Idea

inputs : X', X'' input difference : $\Delta X = X' \oplus X''$

outputs : Y', Y'' output difference : $\Delta Y = Y' \oplus Y''$

differential : $(\Delta X, \Delta Y)$

given $\Delta X^*, \Delta Y^*$, what about $Pr(\Delta Y = \Delta Y^* | \Delta X = \Delta X^*)$? ideally, 2^{-n} (inputs and outputs have n bits)

↓

how to find ? s.t. Pr much $> 2^{-n}$

↓

highly likely differential characteristics

Analyzing the Cipher Components

4×4 S-box :

X	Y	ΔY		
		$\Delta X = 1011$	$\Delta X = 1000$	$\Delta X = 0100$
0000	1110	0010	1101	1100
0001	0100	0010	1110	1011
0010	1101	0111	0101	0110
0011	0001	0010	1011	1001
0100	0010	0101	0111	1100
0101	1111	1111	0110	1011
0110	1011	0010	1011	0110
0111	1000	1101	1111	1001
1000	0011	0010	1101	0110
1001	1010	0111	1110	0011
1010	0110	0010	0101	0110
1011	1100	0010	1011	1011
1100	0101	1101	0111	0110
1101	1001	0010	0110	0011
1110	0000	1111	1011	0110
1111	0111	0101	1111	1011

Table 6. Sample Difference Pairs of the S-box.

e.g. $\Delta X = 1011$, $X = 0000$, ΔY ?

$X' = \Delta X \oplus X = 1011$

$Y' = S(X') = 1100$

$Y = S(X) = 1110$

$\Delta Y = Y' \oplus Y = 0010$

$Pr(\Delta Y = 0010 | \Delta X = 1011) = \frac{8}{16} > 2^{-4}$

$Pr(\Delta Y = 1010 | \Delta X = 0100) = 0$

Difference Distribution Table

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

Table 7. Difference Distribution Table.

calculate :

$DDT_S(\Delta X, \Delta Y)$ = # occurences of ΔY given ΔX

property :

1. $\Sigma_{row} = \Sigma_{column} = 2^n$
2. each element is even ($\Delta X = X' \oplus X'' = X'' \oplus X'$)
3.
4. ideally table : $\begin{matrix} 1 & 1 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & \cdots & 1 \end{matrix}$ → IMPOSSIBLE.

Key Influence on the S-box Differential

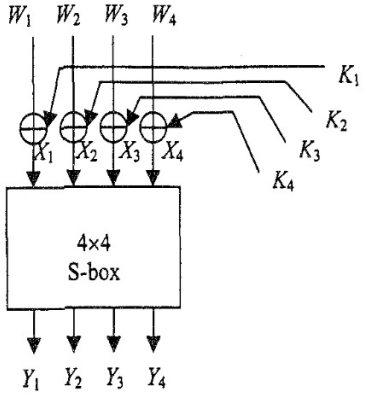

Figure 4. Keyed S-box.

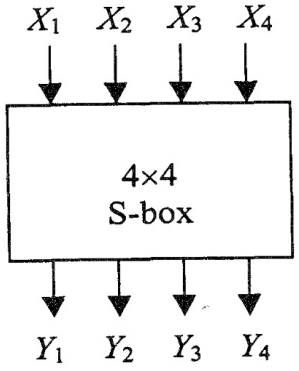

Figure 2. S-box Mapping.

(unkeyed)

$$\begin{aligned} \Delta W &= W' \oplus W'' \\ &= (X' \oplus K) \oplus (X'' \oplus K) \\ &= X' \oplus X'' \\ &= \Delta X \end{aligned}$$

the keyed S-box has the same DDT as the unkeyed S-box.

Constructing Differential Characteristics

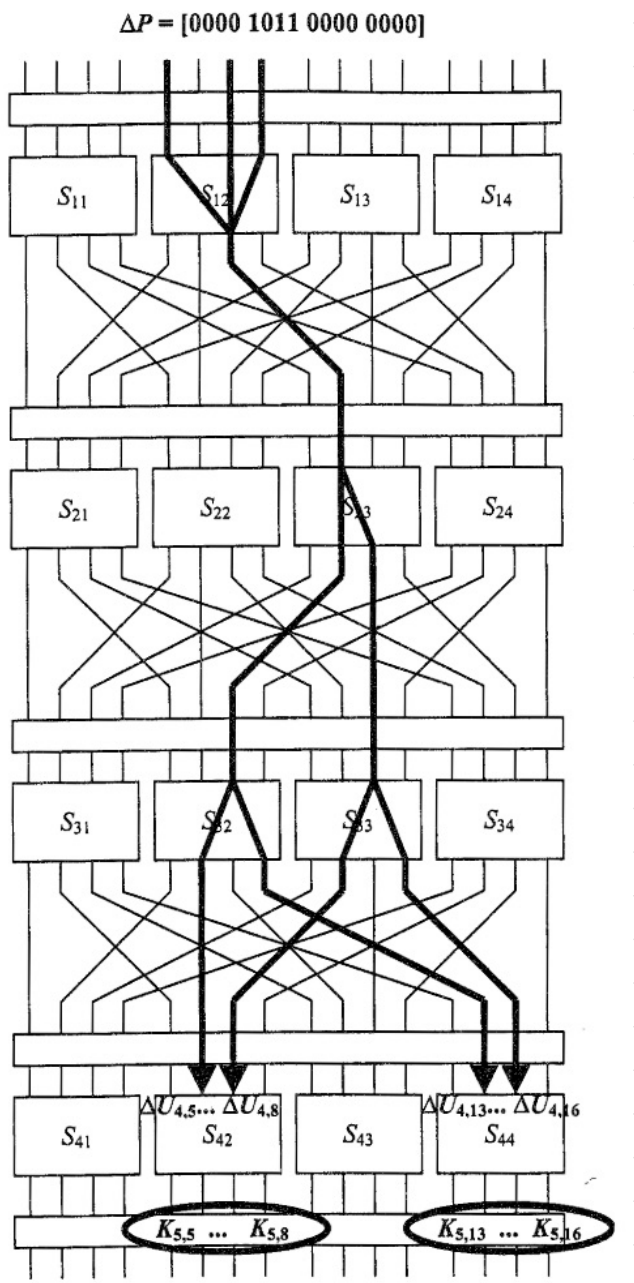

Figure 5. Sample Differential Characteristic.

S_{12} : $\Delta X = B \rightarrow \Delta Y = 2$ $\frac{8}{16}$

S_{23} : $\Delta X = 4 \rightarrow \Delta Y = 6$ $\frac{6}{16}$

S_{32} : $\Delta X = 2 \rightarrow \Delta Y = 5$ $\frac{6}{16}$

S_{33} : $\Delta X = 2 \rightarrow \Delta Y = 5$ $\frac{6}{16}$.

others S-box $\Delta X = 0 \rightarrow \Delta Y = 0$

$$Pr(\Delta V_1 | \Delta P) = \frac{8}{16}$$

$$Pr(\Delta V_2 | \Delta P) = \frac{8}{16} \times \frac{6}{16}$$

$$Pr(\Delta V_3 | \Delta P) = \frac{8}{16} \times \frac{6}{16} \times \left(\frac{6}{16}\right)^2 = \frac{27}{1024}$$

Extracting Key Bits

N_D $(\Delta P, \Delta C)$

⋮ --> ΔV_3

K_5	count
0000 0000 0000 0000	0
0000 0000 0000 0001	0
...	...
0000 1111 0000 1111	0

if $\Delta V_3 == \Delta V_3'$, then ++count[k]

General Algorithm

1. determine a high probability differential characteristic of $R-1$ rounds ← How?
2. collect N_D Δplaintext/Δciphertext samples
3. For all k : count[k] = 0
4. For all k :
   For all (P, C) :
   if $\Delta V_3 == \Delta V_3'$:
   ++count[k]

→ the correct partial subkey will result differences occur frequently.

partial subkey $[K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}]$	prob	partial subkey $[K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}]$	prob
1 C	0.0000	2 A	0.0032
1 D	0.0000	2 B	0.0022
1 E	0.0000	2 C	0.0000
1 F	0.0000	2 D	0.0000
2 0	0.0000	2 E	0.0000
2 1	0.0136	2 F	0.0000
2 2	0.0068	3 0	0.0004
2 3	0.0068	3 1	0.0000
2 4	**0.0244**	3 2	0.0004
2 5	0.0000	3 3	0.0004
2 6	0.0068	3 4	0.0000
2 7	0.0068	3 5	0.0004
2 8	0.0030	3 6	0.0000
2 9	0.0024	3 7	0.0008

Table 8. Experimental Results for Differential Attack.

prob = $\frac{count}{N_D}$

$0.0244 \approx \frac{27}{1024} = 0.0264$

Complexity of the Attack.

the number of active S-boxes ↑

the differential probabilities of the active S-boxes ↓

$\Big\} \Rightarrow$ characteristic probability ↓

$N_D \approx \frac{c}{p_D}$ → c: small constant

→ p_D: differential characteristic probability.

↳ a few occurences of the right pair are enough to give a significantly greater count.

right pairs : $\{(P_1, P_2) \mid P_1 \oplus P_2 = \Delta P\}$

analogous concept of linear hulls.

A Little More to Say

S-boxes play an important role in the security of cipher

↳ mathematically, vectorial boolean functions

↓ need cryptographic properties ⟷ different cryptanalysis

linear, differential, ML-based, ···

special classes of cryptographic Boolean functions

bent functions

APN functions

... ...

[1]Adrien Benamira, David Gerault, Thomas Peyrin, et al. A Deeper Look at Machine Learning-Based Cryptanalysis, 2021.

[2]GOHR A. Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning[M] // Boldyreva, A; Micciancio, D. Advances in Cryptology – CRYPTO 2019. Cham: Springer International Publishing, 2019: 150–179. DOI: 10.1007/978-3-030-26951-7_6.

cryptography	machine learning
learn Dec_k	learn unknown function.
⇒	some learning problems are computationally intractable
cryptanalysis	⇐