LINEAR CRYPTANALYSIS

Samed Bajrić

UNIVERSITY OF PRIMORSKA

FAMNIT

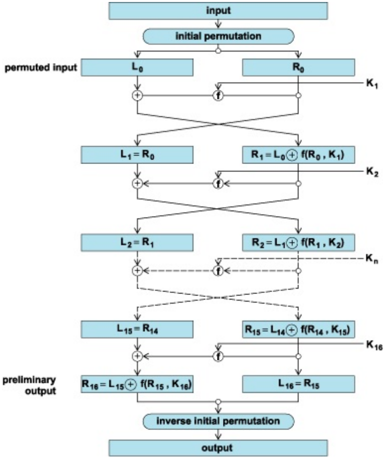
06. June 2011.

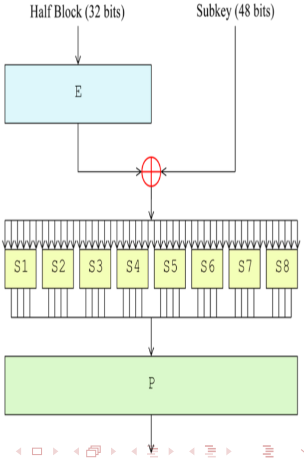
Data Encryption Standard (DES)-history

- The most famous and analyzed block cipher - international banking standard
- Design criteria were kept secret (including differential cryptanalysis) for more than 20 years
- Based on IBM's LUCIFER but changes introduced by NSA
- Still no trapdoors have been found

DES Algorithm

Block length 64 bits, key length 56 bits





Linear Attack on DES idea

In DES S-box maps 6 input bits to 4 output bits i.e.,

$$S : F_2^6 \to F_2^4, \text{ where } F_2 = \{0, 1\}$$

$$(x_1, ..., x_6) \to (y_1, ..., y_4)$$

Assume,

$$y_1 = x_1x_2x_3 + x_2x_5 + x_2 + x_4$$

$$y_2 = x_1x_2x_3 + x_2x_5 + x_1 + x_4$$

$$\Rightarrow y_1 \oplus y_2 = x_1 \oplus x_2$$

Specifically, if P_i are plaintext bits, C_i are ciphertext bits, and K_i are subkey bits, then we wish to find an expression of the form

$$P_{i_1} \oplus P_{i_2} \oplus ...P_{i_j} \oplus C_{i_1} \oplus C_{i_2} \oplus ...C_{i_k} = K_{i_1} \oplus K_{i_2} \oplus ...K_{i_m}$$

such that this expression has a high or low probability of occurrence.

Specifically, if P_i are plaintext bits, C_i are ciphertext bits, and K_i are subkey bits, then we wish to find an expression of the form

$$P_{i_1} \oplus P_{i_2} \oplus ...P_{i_j} \oplus C_{i_1} \oplus C_{i_2} \oplus ...C_{i_k} = K_{i_1} \oplus K_{i_2} \oplus ...K_{i_m}$$

such that this expression has a high or low probability of occurrence.

No such obvious expression should exist, otherwise the cipher is trivially weak. If we were to randomly select bits for the above expression, it would hold exactly $\frac{1}{2}$ the time.

Specifically, if P_i are plaintext bits, C_i are ciphertext bits, and K_i are subkey bits, then we wish to find an expression of the form

$$P_{i_1} \oplus P_{i_2} \oplus ...P_{i_j} \oplus C_{i_1} \oplus C_{i_2} \oplus ...C_{i_k} = K_{i_1} \oplus K_{i_2} \oplus ...K_{i_m}$$

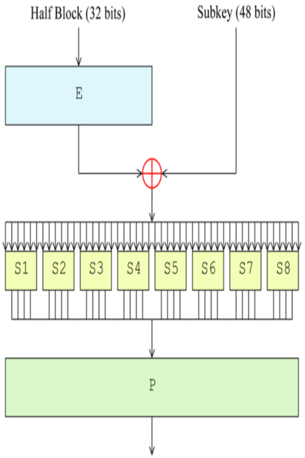
such that this expression has a high or low probability of occurrence.

No such obvious expression should exist, otherwise the cipher is trivially weak. If we were to randomly select bits for the above expression, it would hold exactly $\frac{1}{2}$ the time.

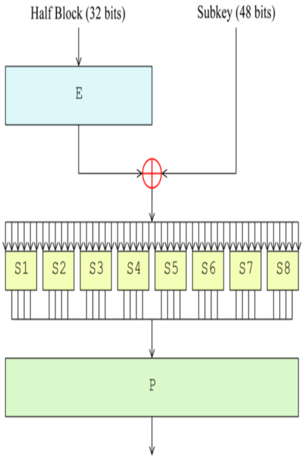
Definition

For a given linear approximation to part of a cipher, let p be the probability that it holds. We refer to $|p - \frac{1}{2}|$ as the **bias** of the approximation.

Reducing DES



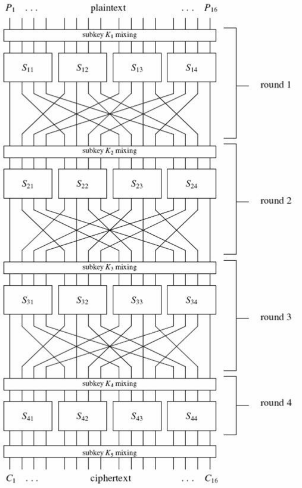
Reducing DES



Let us simplify DES by

- Reducing the code length from 64 to 16
- Removing expansion function $E(R)$
- Taking identical S-boxes of size 4 bits
- Repeating over less rounds

Substitution - Permutation Network



- SPN is a 16-bit block cipher
- Each round consists of a substitution and a permutation
- 4×4-bit S-box
- key-mixing by XOR-ing

SPN S-box and Permutation

SPN uses a single 4-bit S-Box that has the following structure:

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

SPN S-box and Permutation

SPN uses a single 4-bit S-Box that has the following structure:

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

And the following 16-bit permutation:

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

SPN S-box and Permutation

SPN uses a single 4-bit S-Box that has the following structure:

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

And the following 16-bit permutation:

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

The S-Box provides the *confusion* function and the permutation implements the *diffusion* operation in SPN, thus making it cryptographically similar to DES.

Linear and Affine approximation of S-Box

Question:
How do we come up with the desired expression for the entire cipher?

Linear and Affine approximation of S-Box

Question:
How do we come up with the desired expression for the entire cipher?

We start by looking at the only non-linear component, the S-Box.

Linear and Affine approximation of S-Box

Question:
How do we come up with the desired expression for the entire cipher?

We start by looking at the only non-linear component, the S-Box.

To find the linear or affine approximation of the S-Box we simply consider every possible expression of the input bits X_i and output bits Y_j . Thus the expression has the form

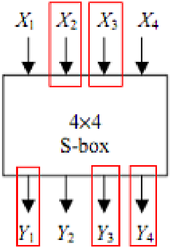
$$\bigoplus_{i \in U} X_i = \bigoplus_{j \in V} Y_j$$

where U and V range over all possible subsets of $\{1, 2, 3, 4\}$. We then compare how often this expression coincides with the S-Box.

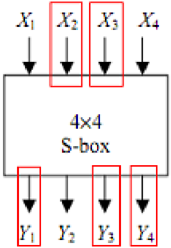
The number of agreements (minus 8) between the S-Box and every possible expression is summarized in the table below. Thus to get the bias, one must only divide by 16.

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

S-Box Approximation Example



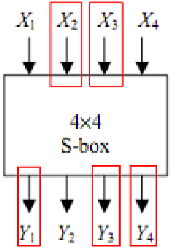
S-Box Approximation Example



We can take the following expression as an example:

$$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$$

S-Box Approximation Example



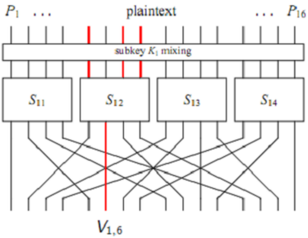
We can take the following expression as an example:

$$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$$

Applying all possible values for the input X bits it turns out that the expression holds in 12 out of the 16 cases. Hence, this expression has a bias of $\frac{12}{16} - \frac{1}{2} = \frac{1}{4}$

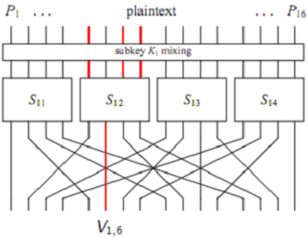
Linear Attack on SPN

What this means for 1 round



Linear Attack on SPN

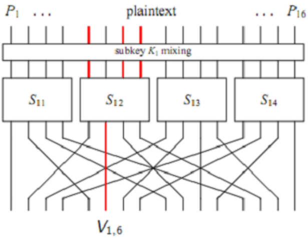
What this means for 1 round



Note, from the previous table, that the expression $X_1 \oplus X_3 \oplus X_4 = Y_2$ has a bias of $+\frac{1}{4}$

Linear Attack on SPN

What this means for 1 round

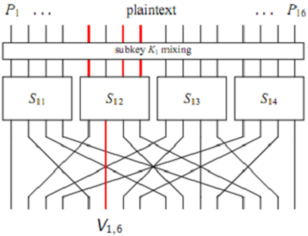


Note, from the previous table, that the expression $X_1 \oplus X_3 \oplus X_4 = Y_2$ has a bias of $+\frac{1}{4}$

Note also that $U_5^1 = P_5 \oplus K_5^1$

Linear Attack on SPN

We can now write down the following linear approximation across the 1st round of SPN:



$$V_6^1 = U_5^1 \oplus U_7^1 \oplus U_8^1$$

$$= (P_5 \oplus K_5^1) \oplus (P_7 \oplus K_7^1) \oplus (P_8 \oplus K_8^1)$$

This expression holds with probability of $\frac{3}{4}$ (bias of $+\frac{1}{4}$)

Linear Attack on SPN

We can get expressions that hold with some non-$\frac{1}{2}$ probability for every round. But we must somehow combine them to write an expression relating the plaintext and ciphertext bits.

Linear Attack on SPN

We can get expressions that hold with some non-$\frac{1}{2}$ probability for every round. But we must somehow combine them to write an expression relating the plaintext and ciphertext bits. We use the *Piling Up Lemma* for this purpose.

Linear Attack on SPN

We can get expressions that hold with some non-$\frac{1}{2}$ probability for every round. But we must somehow combine them to write an expression relating the plaintext and ciphertext bits. We use the *Piling Up Lemma* for this purpose.

Piling - Up Lemma

For n independent, random binary variables $X_1, X_2, ..., X_n$,

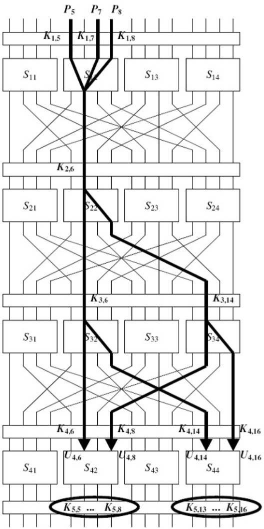
$$Pr(X_1 \oplus X_2 \oplus ... \oplus X_n = 0) = \frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} \varepsilon_i$$

or, equivalently,

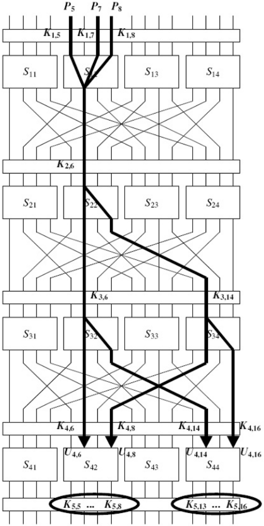
$$\varepsilon_{1,2,...,n} = 2^{n-1} \prod_{i=1}^{n} \varepsilon_i$$

where $\varepsilon_{1,2,...,n}$ represents the bias of $X_1 \oplus X_2 \oplus ... \oplus X_n = 0$

Linear Attack on SPN



Linear Attack on SPN



We can write down 4 approximations

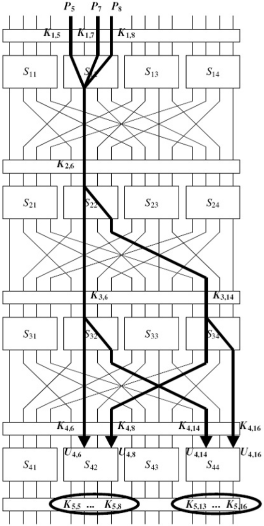
$$S_{12} : X_1 \oplus X_3 \oplus X_4 = Y_2$$

$$S_{22} : X_2 = Y_2 \oplus Y_4$$

$$S_{32} : X_2 = Y_2 \oplus Y_4$$

$$S_{34} : X_2 = Y_2 \oplus Y_4$$

Linear Attack on SPN



We can write down 4 approximations

$$S_{12} : X_1 \oplus X_3 \oplus X_4 = Y_2$$

$$S_{22} : X_2 = Y_2 \oplus Y_4$$

$$S_{32} : X_2 = Y_2 \oplus Y_4$$

$$S_{34} : X_2 = Y_2 \oplus Y_4$$

Each of these has a probability bias magnitude of $\frac{1}{4}$.

Linear Attack on SPN

Consider the first 2 rounds:

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

$$\Rightarrow (P_5 \oplus K_5^1) \oplus (P_7 \oplus K_7^1) \oplus (P_8 \oplus K_8^1) = V_6^1$$

$$X_2 = Y_2 \oplus Y_4 \Rightarrow (V_6^1 \oplus K_6^2 = V_6^2 \oplus V_8^2)$$

Linear Attack on SPN

Consider the first 2 rounds:

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

$$\Rightarrow (P_5 \oplus K_5^1) \oplus (P_7 \oplus K_7^1) \oplus (P_8 \oplus K_8^1) = V_6^1$$

$$X_2 = Y_2 \oplus Y_4 \Rightarrow (V_6^1 \oplus K_6^2 = V_6^2 \oplus V_8^2)$$

Each of these has a bias of magnitude $\frac{1}{4}$ and we can combine to obtain:

$$V_6^2 \oplus V_8^2 \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 = 0$$

By the Piling Up Lemma this holds with bias

$$2 \cdot \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{8}$$

Linear Attack on SPN

Using this principle we can write the following equation over 3 rounds of SPN:

$$U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 \oplus P_5 \oplus P_7 \oplus P_8 = \overline{K}$$

Where
$\overline{K} = K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3 \oplus K_6^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4$.

Note that since the key is fixed, $\overline{K} = 0$ or 1 and thus we can ignore it since we only care about the bias.

Linear Attack on SPN

Using this principle we can write the following equation over 3 rounds of SPN:

$$U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 \oplus P_5 \oplus P_7 \oplus P_8 = \overline{K}$$

Where
$\overline{K} = K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3 \oplus K_6^4 \oplus K_8^4 \oplus K_{14}^4 \oplus K_{16}^4$.

Note that since the key is fixed, $\overline{K} = 0$ or 1 and thus we can ignore it since we only care about the bias.

The magnitude of the bias of the above expression, by the Piling Up Lemma, is $\frac{1}{32}$.

Extracting Key Bits

We can partially undo the last round by guessing the last key.

Extracting Key Bits

We can partially undo the last round by guessing the last key.

If we guess correctly, the equation will hold with high bias. If we guess wrong, the equation will probably hold with probability close to $\frac{1}{2}$, that is, a bias close to 0.

Extracting Key Bits

We can partially undo the last round by guessing the last key.

If we guess correctly, the equation will hold with high bias. If we guess wrong, the equation will probably hold with probability close to $\frac{1}{2}$, that is, a bias close to 0.

BUT to do this, we don't need to guess the entire key for the last round! Our expression only involves 4 fourth round input (U_4) bits, output from 2 S-Boxes of the third round. Thus we only need to guess $2^8 = 256$ values, instead $2^{16} = 65546$, which is huge difference.

Extracting Key Bits

We can partially undo the last round by guessing the last key.

If we guess correctly, the equation will hold with high bias. If we guess wrong, the equation will probably hold with probability close to $\frac{1}{2}$, that is, a bias close to 0.

BUT to do this, we don't need to guess the entire key for the last round! Our expression only involves 4 fourth round input (U_4) bits, output from 2 S-Boxes of the third round. Thus we only need to guess $2^8 = 256$ values, instead $2^{16} = 65546$, which is huge difference.

For each value of the guessed target partial subkey we can undo the last round and determine the bias of the equation. Highest bias indicates likely correct guess.

Conclusion

- One might say that 8 bits of the last round key is not very useful
- How many plaintext/ciphertext pairs do we need to make this attack?

Thank you for your Attention!