Linear Cryptanalysis of SPN

Math 4175

§4.3. Linear Cryptanalysis (Matsui, 1994)

From Oscar's point of view, there are two ways to attack SPN cipher:

§4.3. Linear Cryptanalysis (Matsui, 1994)

From Oscar's point of view, there are two ways to attack SPN cipher:

1. Linear Cryptanalysis
2. Differential Cryptanalysis

§4.3. Linear Cryptanalysis (Matsui, 1994)

From Oscar's point of view, there are two ways to attack SPN cipher:

1. Linear Cryptanalysis
2. Differential Cryptanalysis

In this section, we will outline linear cryptanalysis method by using our previous example of SPN.

§4.3. Linear Cryptanalysis (Matsui, 1994)

From Oscar's point of view, there are two ways to attack SPN cipher:

1. Linear Cryptanalysis
2. Differential Cryptanalysis

In this section, we will outline linear cryptanalysis method by using our previous example of SPN.

Suppose that Oscar knows the type of (SPN) cipher (that is, he knows $\ell = m = N = 4$) and he also has knowledge of a sizable amount of plain texts and the corresponding cipher texts. But he doesn't know which portion of plain texts and cipher texts he holds and has no knowledge of any key.

§4.3. Linear Cryptanalysis

The basic idea is to approximate the operation of a portion of the cipher with a linear expression of the form:

$$X_{i_1} \oplus \cdots \oplus X_{i_r} \oplus U^4_{j_1} \oplus \cdots \oplus U^4_{j_s} = 0 \tag{1}$$

where X_i represents the i-th bit of the input plaintext $[X_1, X_2, \cdots]$ in the first round of the cipher and U^4_j represents j-th bit of the input to the fourth round (last round) of the cipher.

§4.3. Linear Cryptanalysis

The basic idea is to approximate the operation of a portion of the cipher with a linear expression of the form:

$$X_{i_1} \oplus \cdots \oplus X_{i_r} \oplus U^4_{j_1} \oplus \cdots \oplus U^4_{j_s} = 0 \tag{1}$$

where X_i represents the i-th bit of the input plaintext $[X_1, X_2, \cdots]$ in the first round of the cipher and U^4_j represents j-th bit of the input to the fourth round (last round) of the cipher.

In a perfectly random scenario, we get $Pr[L = 0] = 1/2$ and $Pr[L = 1] = 1/2$ where L is the left side of the above expression (1).

§4.3. Linear Cryptanalysis

The basic idea is to approximate the operation of a portion of the cipher with a linear expression of the form:

$$X_{i_1} \oplus \cdots \oplus X_{i_r} \oplus U^4_{j_1} \oplus \cdots \oplus U^4_{j_s} = 0 \tag{1}$$

where X_i represents the i-th bit of the input plaintext $[X_1, X_2, \cdots]$ in the first round of the cipher and U^4_j represents j-th bit of the input to the fourth round (last round) of the cipher.

In a perfectly random scenario, we get $Pr[L = 0] = 1/2$ and $Pr[L = 1] = 1/2$ where L is the left side of the above expression (1).

Oscar's goal is to exploit by finding such expressions with probabilities away from $1/2$. For this purpose, we need to establish some more results in probability theory.

§4.3. Piling-Up Lemma

Suppose that $X_1, X_2, \cdots$ are independent random variables taking on values from the set $\{0, 1\}$ (binary) with probabilities:

$$Pr[X_i = 0] = p_i \text{ and hence } Pr[X_i = 1] = 1 - p_i.$$

§4.3. Piling-Up Lemma

Suppose that $X_1, X_2, \cdots$ are independent random variables taking on values from the set $\{0, 1\}$ (binary) with probabilities:

$$Pr[X_i = 0] = p_i \text{ and hence } Pr[X_i = 1] = 1 - p_i.$$

The independence of X_i and X_j (for $i \neq j$) implies:

$$\begin{aligned}
Pr[X_i = 0, X_j = 0] &= p_i p_j \\
Pr[X_i = 0, X_j = 1] &= p_i(1 - p_j) \\
Pr[X_i = 1, X_j = 0] &= (1 - p_i)p_j \\
Pr[X_i = 1, X_j = 1] &= (1 - p_i)(1 - p_j)
\end{aligned}$$

§4.3. Piling-Up Lemma

Now consider the discrete random variable $X_i \oplus X_j$, which is $X_i + X_j$ mod 2.

§4.3. Piling-Up Lemma

Now consider the discrete random variable $X_i \oplus X_j$, which is $X_i + X_j$ mod 2. We have

$$\begin{aligned}
Pr[X_i \oplus X_j = 0] &= Pr[X_i = 0, X_j = 0] + Pr[X_i = 1, X_j = 1] \\
&= p_i p_j + (1 - p_i)(1 - p_j) \\
Pr[X_i \oplus X_j = 1] &= Pr[X_i = 0, X_j = 1] + Pr[X_i = 1, X_j = 0] \\
&= p_i(1 - p_j) + (1 - p_i)p_j
\end{aligned}$$

§4.3. Piling-Up Lemma

Now consider the discrete random variable $X_i \oplus X_j$, which is $X_i + X_j$ mod 2. We have

$$\begin{aligned} Pr[X_i \oplus X_j = 0] &= Pr[X_i = 0, X_j = 0] + Pr[X_i = 1, X_j = 1] \\ &= p_i p_j + (1 - p_i)(1 - p_j) \\ Pr[X_i \oplus X_j = 1] &= Pr[X_i = 0, X_j = 1] + Pr[X_i = 1, X_j = 0] \\ &= p_i(1 - p_j) + (1 - p_i)p_j \end{aligned}$$

For our purpose, we are interested in probabilities away from $1/2$ and so it is often convenient to express above formulas in terms of a quantity called the bias.

§4.3. Piling-Up Lemma

Now consider the discrete random variable $X_i \oplus X_j$, which is $X_i + X_j$ mod 2. We have

$$\begin{aligned} Pr[X_i \oplus X_j = 0] &= Pr[X_i = 0, X_j = 0] + Pr[X_i = 1, X_j = 1] \\ &= p_i p_j + (1 - p_i)(1 - p_j) \\ Pr[X_i \oplus X_j = 1] &= Pr[X_i = 0, X_j = 1] + Pr[X_i = 1, X_j = 0] \\ &= p_i(1 - p_j) + (1 - p_i)p_j \end{aligned}$$

For our purpose, we are interested in probabilities away from $1/2$ and so it is often convenient to express above formulas in terms of a quantity called the bias.

Definition: The bias of X_i is defined to be the quantity

$$\epsilon_i = p_i - 1/2 \text{ or } p_i = \epsilon_i + 1/2.$$

§4.3. Piling-Up Lemma

Observe that for $i = 1, 2, \cdots$, we have:

$$-0.5 \leq \epsilon_i \leq 0.5,$$

$$Pr[X_i = 0] = p_i = 0.5 + \epsilon_i, \text{ and}$$

$$Pr[X_i = 1] = 1 - p_i = 0.5 - \epsilon_i$$

§4.3. Piling-Up Lemma

Observe that for $i = 1, 2, \cdots$, we have:

$$-0.5 \leq \epsilon_i \leq 0.5,$$

$$Pr[X_i = 0] = p_i = 0.5 + \epsilon_i, \text{ and}$$
$$Pr[X_i = 1] = 1 - p_i = 0.5 - \epsilon_i$$

Notice that

$$\begin{aligned} Pr[X_i \oplus X_j = 0] &= (0.5 + \epsilon_i)(0.5 + \epsilon_j) + (0.5 - \epsilon_i)(0.5 - \epsilon_j) \\ &= 0.5 + 2\epsilon_i\epsilon_j \end{aligned}$$

So the bias of $X_i \oplus X_j$, denoted by $\epsilon_{i,j}$ is equal to $2\epsilon_i\epsilon_j$.

§4.3. Piling-Up Lemma

Piling-Up Lemma: Let $\epsilon_{1,2,\cdots k}$ be the bias of the random variable $X_1 \oplus \cdots \oplus X_k$. Then

$$\epsilon_{1,2,\cdots k} = 2^{k-1} \prod_{j=1}^{k} \epsilon_j$$

§4.3. Piling-Up Lemma

Piling-Up Lemma: Let $\epsilon_{1,2,\cdots k}$ be the bias of the random variable $X_1 \oplus \cdots \oplus X_k$. Then

$$\epsilon_{1,2,\cdots k} = 2^{k-1} \prod_{j=1}^{k} \epsilon_j$$

Proof: The proof follows by induction on k.

§4.3. Piling-Up Lemma

Piling-Up Lemma: Let $\epsilon_{1,2,\cdots k}$ be the bias of the random variable $X_1 \oplus \cdots \oplus X_k$. Then

$$\epsilon_{1,2,\cdots k} = 2^{k-1} \prod_{j=1}^{k} \epsilon_j$$

Proof: The proof follows by induction on k.

For $k = 1$, the result holds obviously.

§4.3. Piling-Up Lemma

Piling-Up Lemma: Let $\epsilon_{1,2,\cdots k}$ be the bias of the random variable $X_1 \oplus \cdots \oplus X_k$. Then

$$\epsilon_{1,2,\cdots k} = 2^{k-1} \prod_{j=1}^{k} \epsilon_j$$

Proof: The proof follows by induction on k.

For $k = 1$, the result holds obviously.

For $k = 2$: notice from the previous slide that $Pr[X_1 \oplus X_2 = 0] = 0.5 + 2\epsilon_1\epsilon_2$.

§4.3. Piling-Up Lemma

Piling-Up Lemma: Let $\epsilon_{1,2,\cdots k}$ be the bias of the random variable $X_1 \oplus \cdots \oplus X_k$. Then

$$\epsilon_{1,2,\cdots k} = 2^{k-1} \prod_{j=1}^{k} \epsilon_j$$

Proof: The proof follows by induction on k.

For $k = 1$, the result holds obviously.

For $k = 2$: notice from the previous slide that
$Pr[X_1 \oplus X_2 = 0] = 0.5 + 2\epsilon_1\epsilon_2$.

Hence the bias $\epsilon_{1,2} = (0.5 + 2\epsilon_1\epsilon_2) - 0.5 = 2\epsilon_1\epsilon_2$.

§4.3. Piling-Up Lemma

Suppose that the result holds for $k = \ell$, where $\ell \geq 2$. We will then prove that the result holds for $k = \ell + 1$.

§4.3. Piling-Up Lemma

Suppose that the result holds for $k = \ell$, where $\ell \geq 2$. We will then prove that the result holds for $k = \ell + 1$.

That is, we want to find the bias of $X_1 \oplus \cdots \oplus X_{\ell+1}$.

§4.3. Piling-Up Lemma

Suppose that the result holds for $k = \ell$, where $\ell \geq 2$. We will then prove that the result holds for $k = \ell + 1$.

That is, we want to find the bias of $X_1 \oplus \cdots \oplus X_{\ell+1}$.

Let us split the above random variable into two parts as follows:
$X_1 \oplus \cdots \oplus X_{\ell+1} = (X_1 \oplus \cdots \oplus X_\ell) \oplus X_{\ell+1}$.

§4.3. Piling-Up Lemma

Suppose that the result holds for $k = \ell$, where $\ell \geq 2$. We will then prove that the result holds for $k = \ell + 1$.

That is, we want to find the bias of $X_1 \oplus \cdots \oplus X_{\ell+1}$.

Let us split the above random variable into two parts as follows:
$X_1 \oplus \cdots \oplus X_{\ell+1} = (X_1 \oplus \cdots \oplus X_\ell) \oplus X_{\ell+1}$.

By using the induction and the formula for $k = 2$, we see that the bias of $X_1 \oplus \cdots \oplus X_{\ell+1}$ is

$$2 \times \left(2^{\ell-1} \prod_{j=1}^{\ell} \epsilon_j\right) \times \epsilon_{\ell+1} = 2^{\ell} \prod_{j=1}^{\ell+1} \epsilon_j$$

§4.3. Piling-Up Lemma

Suppose that the result holds for $k = \ell$, where $\ell \geq 2$. We will then prove that the result holds for $k = \ell + 1$.

That is, we want to find the bias of $X_1 \oplus \cdots \oplus X_{\ell+1}$.

Let us split the above random variable into two parts as follows: $X_1 \oplus \cdots \oplus X_{\ell+1} = (X_1 \oplus \cdots \oplus X_\ell) \oplus X_{\ell+1}$.

By using the induction and the formula for $k = 2$, we see that the bias of $X_1 \oplus \cdots \oplus X_{\ell+1}$ is

$$2 \times \left(2^{\ell-1} \prod_{j=1}^{\ell} \epsilon_j\right) \times \epsilon_{\ell+1} = 2^\ell \prod_{j=1}^{\ell+1} \epsilon_j$$

Now the proof is complete by induction.

§4.3. Piling-Up Lemma

Corollary: The bias of $X_1 \oplus \cdots \oplus X_k$ is zero if the bias of X_j is zero for some j such that $1 \leq j \leq k$.

§4.3. Piling-Up Lemma

Corollary: The bias of $X_1 \oplus \cdots \oplus X_k$ is zero if the bias of X_j is zero for some j such that $1 \leq j \leq k$.

Remarks:

1. Piling-Up lemma holds, in general, only when the relevant random variables are independent.

§4.3. Piling-Up Lemma

Corollary: The bias of $X_1 \oplus \cdots \oplus X_k$ is zero if the bias of X_j is zero for some j such that $1 \leq j \leq k$.

Remarks:

1. Piling-Up lemma holds, in general, only when the relevant random variables are independent.

2. Recall that Oscar would like to exploit by constructing linear expression $L = 0$ (where L is as given in (1) in a previous slide) with $Pr[L = 0]$ away from 0.5, or in other words, with non-zero bias.

§4.3. Piling-Up Lemma

Corollary: The bias of $X_1 \oplus \cdots \oplus X_k$ is zero if the bias of X_j is zero for some j such that $1 \leq j \leq k$.

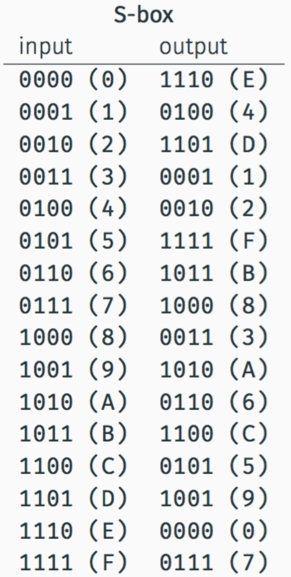
Remarks:

1. Piling-Up lemma holds, in general, only when the relevant random variables are independent.

2. Recall that Oscar would like to exploit by constructing linear expression $L = 0$ (where L is as given in (1) in a previous slide) with $Pr[L = 0]$ away from 0.5, or in other words, with non-zero bias.

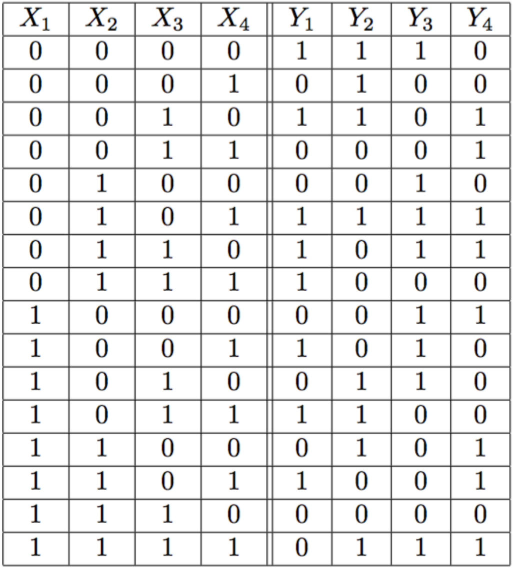
3. This construction is achieved by considering the properties of cipher's only non-linear component: the S-box.

§4.3. S-Box



S-box

input	output
0000 (0)	1110 (E)
0001 (1)	0100 (4)
0010 (2)	1101 (D)
0011 (3)	0001 (1)
0100 (4)	0010 (2)
0101 (5)	1111 (F)
0110 (6)	1011 (B)
0111 (7)	1000 (8)
1000 (8)	0011 (3)
1001 (9)	1010 (A)
1010 (A)	0110 (6)
1011 (B)	1100 (C)
1100 (C)	0101 (5)
1101 (D)	1001 (9)
1110 (E)	0000 (0)
1111 (F)	0111 (7)



X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0
0	0	1	0	1	1	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	1	0	0
1	1	0	0	0	1	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	0	0	0
1	1	1	1	0	1	1	1

Table 1

§4.3. Linear Approximations of S-Box

We recorded the possible values taken by the eight random variables $X_1, \ldots, X_4, Y_1, \ldots, Y_4$ for our S-box in the rows of Table 1.

§4.3. Linear Approximations of S-Box

We recorded the possible values taken by the eight random variables $X_1, \ldots, X_4, Y_1, \ldots, Y_4$ for our S-box in the rows of Table 1.

For example, now let us compute:

$$Pr[X_1 \oplus X_4 \oplus Y_2 = 0]$$

for the S-box used in our cipher.

§4.3. Linear Approximations of S-Box

We recorded the possible values taken by the eight random variables $X_1, \ldots, X_4, Y_1, \ldots, Y_4$ for our S-box in the rows of Table 1.

For example, now let us compute:

$$Pr[X_1 \oplus X_4 \oplus Y_2 = 0]$$

for the S-box used in our cipher.

That is, consider the linear equation $X_1 \oplus X_4 \oplus Y_2 = 0$, or equivalently, $X_1 \oplus X_4 = Y_2$.

§4.3. Linear Approximations of S-Box

We recorded the possible values taken by the eight random variables $X_1, \dots, X_4, Y_1, \dots, Y_4$ for our S-box in the rows of Table 1.

For example, now let us compute:

$$Pr[X_1 \oplus X_4 \oplus Y_2 = 0]$$

for the S-box used in our cipher.

That is, consider the linear equation $X_1 \oplus X_4 \oplus Y_2 = 0$, or equivalently, $X_1 \oplus X_4 = Y_2$.

So we can count the number of rows in the above table in which $X_1 \oplus X_4 = Y_2$ as given in the next slide.

§4.3. Linear Approximations of S-Box

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	$X_1 \oplus X_4$	Y_2	
0	0	0	0	1	1	1	0	0	1	No
0	0	0	1	0	1	0	0	1	1	Yes
0	0	1	0	1	1	0	1	0	1	No
0	0	1	1	0	0	0	1	1	0	No
0	1	0	0	0	0	1	0	0	0	Yes
0	1	0	1	1	1	1	1	1	1	Yes
0	1	1	0	1	0	1	1	0	0	Yes
0	1	1	1	1	0	0	0	1	0	No
1	0	0	0	0	0	1	1	1	0	No
1	0	0	1	1	0	1	0	0	0	Yes
1	0	1	0	0	1	1	0	1	1	Yes
1	0	1	1	1	1	0	0	0	1	No
1	1	0	0	0	1	0	1	1	1	Yes
1	1	0	1	1	0	0	1	0	0	Yes
1	1	1	0	0	0	0	0	1	0	No
1	1	1	1	0	1	1	1	0	1	No

§4.3. Linear Approximations of S-Box

Now $X_1 \oplus X_4$ can be represented by $(1, 0, 0, 1)$, this is called as input sum, which is 9 in hexadecimal; similarly for Y_2, the output sum is $(0, 1, 0, 0)$, which is 4 in hexadecimal.

§4.3. Linear Approximations of S-Box

Now $X_1 \oplus X_4$ can be represented by $(1, 0, 0, 1)$, this is called as input sum, which is 9 in hexadecimal; similarly for Y_2, the output sum is $(0, 1, 0, 0)$, which is 4 in hexadecimal.

The number of "Yeses" in the previous table, which is 8, is denoted by $N_L(9, 4)$, that is,

$$N_L(9, 4) = 8$$

and then dividing by 16 (which is the total number of rows in the table), it is easily seen that

$$Pr[X_1 \oplus X_4 \oplus Y_2 = 0] = \frac{N_L(9, 4)}{16} = \frac{1}{2}$$

§4.3. Linear Approximations of S-Box

Now $X_1 \oplus X_4$ can be represented by $(1, 0, 0, 1)$, this is called as input sum, which is 9 in hexadecimal; similarly for Y_2, the output sum is $(0, 1, 0, 0)$, which is 4 in hexadecimal.

The number of "Yeses" in the previous table, which is 8, is denoted by $N_L(9, 4)$, that is,

$$N_L(9,4) = 8$$

and then dividing by 16 (which is the total number of rows in the table), it is easily seen that

$$Pr[X_1 \oplus X_4 \oplus Y_2 = 0] = \frac{N_L(9,4)}{16} = \frac{1}{2}$$

Hence, the bias of the random variable $X_1 \oplus X_4 \oplus Y_2$ is zero. But we are looking for the linear expression with non-zero bias, that is, the probability away from 0.5.

§4.3. Linear Approximations of S-Box

Now let us analyze the linear equation $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4 = 0$, or equivalently, $X_3 \oplus X_4 = Y_1 \oplus Y_4$.

§4.3. Linear Approximations of S-Box

Now let us analyze the linear equation $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4 = 0$, or equivalently, $X_3 \oplus X_4 = Y_1 \oplus Y_4$.

So now the input sum is $(0, 0, 1, 1)$, which is 3 in hexadecimal notation.

§4.3. Linear Approximations of S-Box

Now let us analyze the linear equation $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4 = 0$, or equivalently, $X_3 \oplus X_4 = Y_1 \oplus Y_4$.

So now the input sum is $(0, 0, 1, 1)$, which is 3 in hexadecimal notation.

The output sum is $(1, 0, 0, 1)$, which is 9 in hexadecimal notation.

§4.3. Linear Approximations of S-Box

Now let us analyze the linear equation $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4 = 0$, or equivalently, $X_3 \oplus X_4 = Y_1 \oplus Y_4$.

So now the input sum is $(0, 0, 1, 1)$, which is 3 in hexadecimal notation.

The output sum is $(1, 0, 0, 1)$, which is 9 in hexadecimal notation.

We see, by counting the number of "Yeses" from the table in the next slide, that

$$N_L(3, 9) = 2$$

§4.3. Linear Approximations of S-Box

Now let us analyze the linear equation $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4 = 0$, or equivalently, $X_3 \oplus X_4 = Y_1 \oplus Y_4$.

So now the input sum is $(0, 0, 1, 1)$, which is 3 in hexadecimal notation.

The output sum is $(1, 0, 0, 1)$, which is 9 in hexadecimal notation.

We see, by counting the number of "Yeses" from the table in the next slide, that

$$N_L(3, 9) = 2$$

$$Pr[X_3 \oplus X_4 \oplus Y_1 \oplus Y_4 = 0] = \frac{N_L(3,9)}{16} = \frac{1}{8}.$$

So we find that the bias of $X_3 \oplus X_4 \oplus Y_1 \oplus Y_4$ is

$$\epsilon(3, 9) = \frac{N_L(3,9)}{16} - \frac{1}{2} = \frac{N_L(3,9) - 8}{16} = \frac{2-8}{16} = -\frac{3}{8}.$$

§4.3. Linear Approximations of S-Box

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	$X_3 \oplus X_4$	$Y_1 \oplus Y_4$	
0	0	0	0	1	1	1	0	0	1	No
0	0	0	1	0	1	0	0	1	0	No
0	0	1	0	1	1	0	1	1	0	No
0	0	1	1	0	0	0	1	0	1	No
0	1	0	0	0	0	1	0	0	0	Yes
0	1	0	1	1	1	1	1	1	0	No
0	1	1	0	1	0	1	1	1	0	No
0	1	1	1	1	0	0	0	0	1	No
1	0	0	0	0	0	1	1	0	1	No
1	0	0	1	1	0	1	0	1	1	Yes
1	0	1	0	0	1	1	0	1	0	No
1	0	1	1	1	1	0	0	0	1	No
1	1	0	0	0	1	0	1	0	1	No
1	1	0	1	1	0	0	1	1	0	No
1	1	1	0	0	0	0	0	1	0	No
1	1	1	1	0	1	1	1	0	1	No

§4.3. Linear Approximations of S-Box

Indeed it is not difficult to compute the biases of all $2^8 = 256$ possible random variables of this form.

§4.3. Linear Approximations of S-Box

Indeed it is not difficult to compute the biases of all $2^8 = 256$ possible random variables of this form.

Then in order to have a compact notation, for a random variable having (hexadecimal) input sum $a = (a_1, a_2, a_3, a_4)$ and $b = (b_1, b_2, b_3, b_4)$, in binary, let $N_L(a, b)$ denote the number of "Yeses" in the corresponding table such that

$$(y_1, y_2, y_3, y_4) = \pi_S(x_1, x_2, x_3, x_4) \qquad \text{and} \qquad \left(\bigoplus_{i=1}^{4} a_i X_i\right) = \left(\bigoplus_{i=1}^{4} b_i Y_i\right)$$

§4.3. Linear Approximations of S-Box

Indeed it is not difficult to compute the biases of all $2^8 = 256$ possible random variables of this form.

Then in order to have a compact notation, for a random variable having (hexadecimal) input sum $a = (a_1, a_2, a_3, a_4)$ and $b = (b_1, b_2, b_3, b_4)$, in binary, let $N_L(a, b)$ denote the number of "Yeses" in the corresponding table such that

$$(y_1, y_2, y_3, y_4) = \pi_S(x_1, x_2, x_3, x_4) \qquad \text{and} \qquad \left(\bigoplus_{i=1}^{4} a_i X_i\right) = \left(\bigoplus_{i=1}^{4} b_i Y_i\right)$$

Then the bias of the random variable having input sum a and output sum b is computed as

$$\epsilon(a, b) = \frac{N_L(a, b) - 8}{16}.$$

§4.3. Linear Approximations of S-Box

Indeed it is not difficult to compute the biases of all $2^8 = 256$ possible random variables of this form.

Then in order to have a compact notation, for a random variable having (hexadecimal) input sum $a = (a_1, a_2, a_3, a_4)$ and $b = (b_1, b_2, b_3, b_4)$, in binary, let $N_L(a, b)$ denote the number of "Yeses" in the corresponding table such that

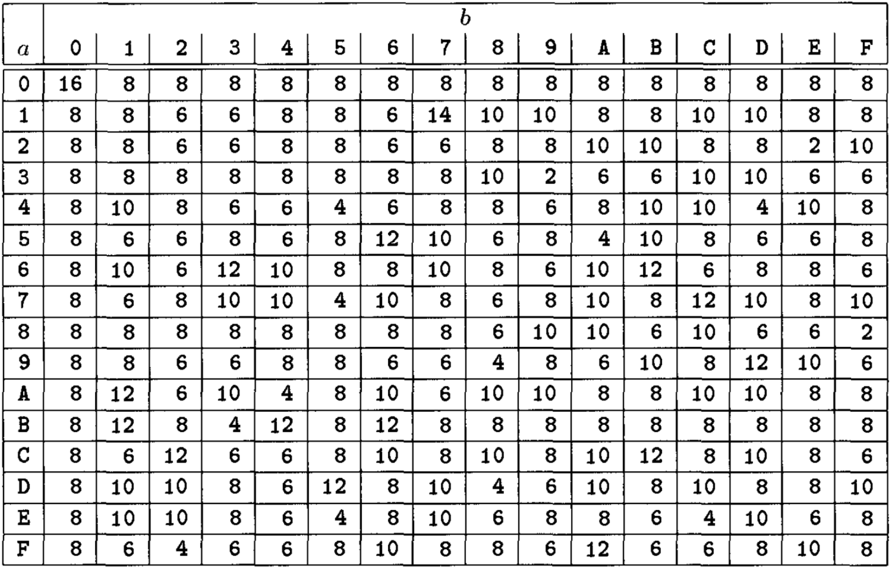
$$(y_1, y_2, y_3, y_4) = \pi_S(x_1, x_2, x_3, x_4) \qquad \text{and} \qquad \left(\bigoplus_{i=1}^{4} a_i X_i\right) = \left(\bigoplus_{i=1}^{4} b_i Y_i\right)$$

Then the bias of the random variable having input sum a and output sum b is computed as

$$\epsilon(a, b) = \frac{N_L(a, b) - 8}{16}.$$

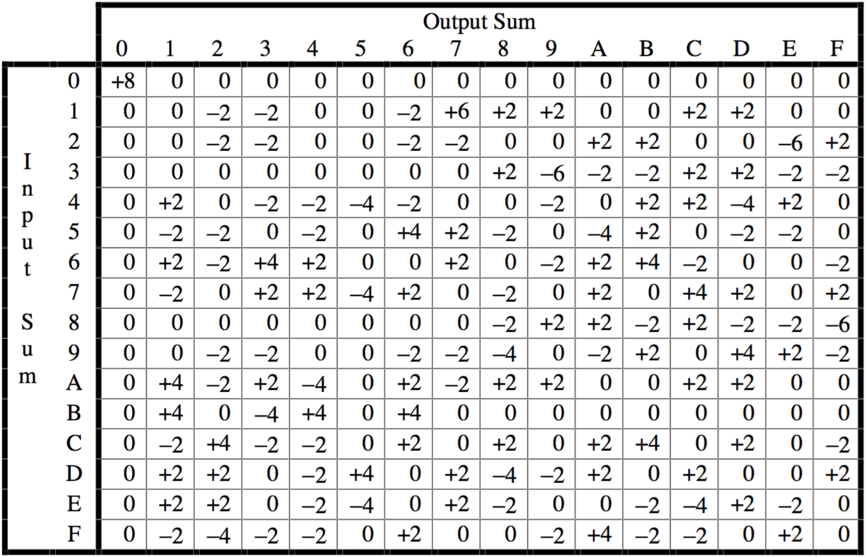
For example, we computed that $N_L(9, 4) = 8$, and hence $\epsilon(9, 4) = 0$.

§4.3. Linear Approximation Table of S-Box: $N_L(a, b)$



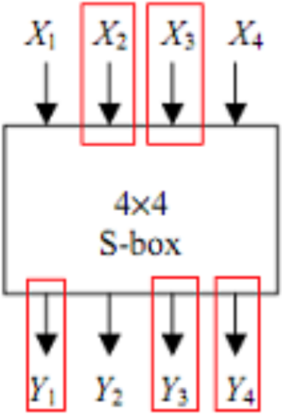
	b															
a	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10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

Normalized Linear Approx. Table of S-Box: $N_L(a, b) - 8$



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

§4.3 S-Box Approximation Example



We can take, as an example, the linear approximation: $X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$, i.e. $(X_2 \oplus X_3) \oplus (Y_1 \oplus Y_3 \oplus Y_4) = 0$. This corresponds to the random variable $(X_2 \oplus X_3) \oplus (Y_1 \oplus Y_3 \oplus Y_4)$ whose bias is given by

$$N_L(6, B) - 8 = 4 \text{ and}$$

$$\epsilon(6, B) = 4/16 = 1/4$$

§4.3 Construction of Linear Approximations

Goal of Oscar: Find a linear equation of the form (1) as given in slide 3 between the input bits X_i's and the bits at U^4 level with nonzero bias.

§4.3 Construction of Linear Approximations

Goal of Oscar: Find a linear equation of the form (1) as given in slide 3 between the input bits X_i's and the bits at U^4 level with nonzero bias.

Linear cryptanalysis requires finding 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).

§4.3 Construction of Linear Approximations

Goal of Oscar: Find a linear equation of the form (1) as given in slide 3 between the input bits X_i's and the bits at U^4 level with nonzero bias.

Linear cryptanalysis requires finding 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).

In other words, this can be achieved by concatenating appropriate linear approximations of S-boxes in successive rounds. We illustrate this process by using our SPN cipher given earlier as an example.

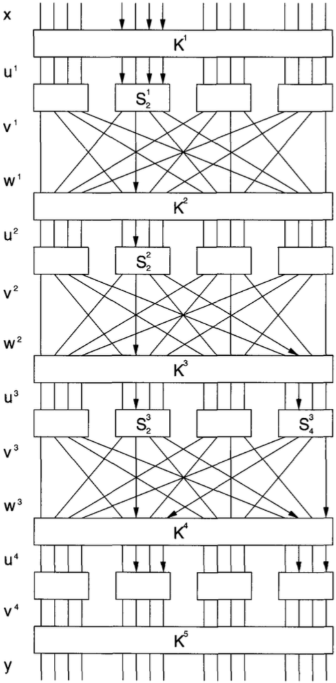
§4.3 Construction of Linear Approximations

Goal of Oscar: Find a linear equation of the form (1) as given in slide 3 between the input bits X_i's and the bits at U^4 level with nonzero bias.

Linear cryptanalysis requires finding 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).

In other words, this can be achieved by concatenating appropriate linear approximations of S-boxes in successive rounds. We illustrate this process by using our SPN cipher given earlier as an example.

The diagram in the next slide can be interpreted as follows: Lines with arrows correspond to random variables which will be involved in linear approximations. The labeled S-boxes are the ones used in these approximations (they are called active S-boxes).



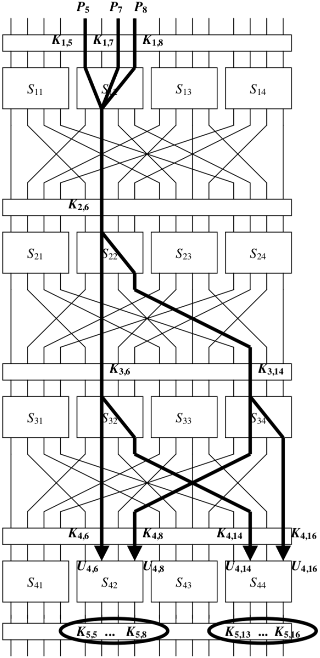
Good linear approximations should have high bias in magnitude. Inputs and/or outputs are spare (involves only 1 or 2 random variables).

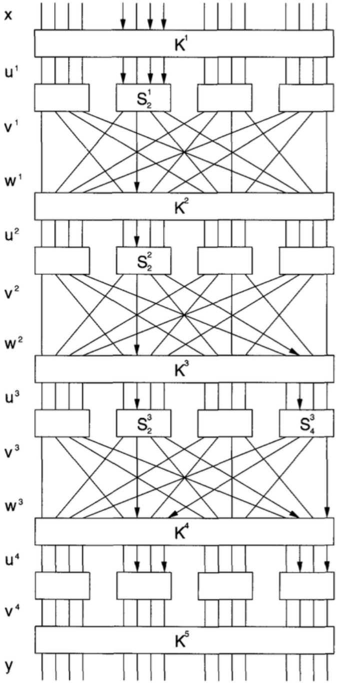
Use table to choose $(B, 4)$ for high bias with maximum active input bits and minimum active output bits.

- S_2^1: $X_1 \oplus X_3 \oplus X_4 = Y_2$

Use table to choose $(4, 5)$ for high bias with minimum active output bits.

- S_2^2: $X_2 = Y_2 \oplus Y_4$
- S_2^3: $X_2 = Y_2 \oplus Y_4$
- S_4^3: $X_2 = Y_2 \oplus Y_4$


P_5
P_7
P_8
$K_{1,5}$
$K_{1,7}$
$K_{1,8}$
S_{11}
S_{12}
S_{13}
S_{14}
$K_{2,6}$
S_{21}
S_{22}
S_{23}
S_{24}
$K_{3,6}$
$K_{3,14}$
S_{31}
S_{32}
S_{33}
S_{34}
$K_{4,6}$
$K_{4,8}$
$K_{4,14}$
$K_{4,16}$
$U_{4,6}$
$U_{4,8}$
$U_{4,14}$
$U_{4,16}$
S_{41}
S_{42}
S_{43}
S_{44}
$K_{5,5}$ ... $K_{5,8}$
$K_{5,13}$ ... $K_{5,16}$



- Now let us connect the trail.
- In S_2^1, the random variable $T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1$ has bias is $1/4$.
- In S_2^2, the random variable $T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2$ has bias -$1/4$.
- In S_2^3, the random variable $T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3$ has bias -$1/4$.
- In S_4^3, the random variable $T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3$ has bias -$1/4$.

§4.3 A Linear Attack on an SPN

T_1, T_2, T_3, T_4 have biases that are relatively high in absolute value.

§4.3 A Linear Attack on an SPN

T_1, T_2, T_3, T_4 have biases that are relatively high in absolute value.

By assuming that these four random variables are independent, we compute the bias of their x-or using the piling up lemma.

§4.3 A Linear Attack on an SPN

T_1, T_2, T_3, T_4 have biases that are relatively high in absolute value.

By assuming that these four random variables are independent, we compute the bias of their x-or using the piling up lemma.

We therefore hypothesize that the random variable

$$T_1 \oplus T_2 \oplus T_3 \oplus T_4$$

has bias equal to $2^3(1/4)(-1/4)^3 = -1/32$.

§4.3 A Linear Attack on an SPN

T_1, T_2, T_3, T_4 have biases that are relatively high in absolute value.

By assuming that these four random variables are independent, we compute the bias of their x-or using the piling up lemma.

We therefore hypothesize that the random variable

$$T_1 \oplus T_2 \oplus T_3 \oplus T_4$$

has bias equal to $2^3(1/4)(-1/4)^3 = -1/32$.

Though the assumption of independence in the above approximation of S-boxes is not strictly correct, it works well, in general, for most ciphers.

§4.3 A Linear Attack on an SPN

The random variables T_1, T_2, T_3 and T_4 have the property that their x-or can be expressed in terms of plaintext bits, bits of U^4 (input to the last round of S-boxes) and key bits as follows:

§4.3 A Linear Attack on an SPN

The random variables T_1, T_2, T_3 and T_4 have the property that their x-or can be expressed in terms of plaintext bits, bits of U^4 (input to the last round of S-boxes) and key bits as follows:

$$T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1 = (P_5 \oplus K_5^1) \oplus (P_7 \oplus K_7^1) \oplus (P_8 \oplus K_8^1) \oplus V_6^1$$

$$T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2 = (V_6^1 \oplus K_6^2) \oplus V_6^2 \oplus V_8^2$$

$$T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3 = (V_6^2 \oplus K_6^3) \oplus V_6^3 \oplus V_8^3$$

$$T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3 = (V_8^2 \oplus K_{14}^3) \oplus V_{14}^3 \oplus V_{16}^3$$

§4.3 A Linear Attack on an SPN

The random variables T_1, T_2, T_3 and T_4 have the property that their x-or can be expressed in terms of plaintext bits, bits of U^4 (input to the last round of S-boxes) and key bits as follows:

$$T_1 = U_5^1 \oplus U_7^1 \oplus U_8^1 \oplus V_6^1 = (P_5 \oplus K_5^1) \oplus (P_7 \oplus K_7^1) \oplus (P_8 \oplus K_8^1) \oplus V_6^1$$

$$T_2 = U_6^2 \oplus V_6^2 \oplus V_8^2 = (V_6^1 \oplus K_6^2) \oplus V_6^2 \oplus V_8^2$$

$$T_3 = U_6^3 \oplus V_6^3 \oplus V_8^3 = (V_6^2 \oplus K_6^3) \oplus V_6^3 \oplus V_8^3$$

$$T_4 = U_{14}^3 \oplus V_{14}^3 \oplus V_{16}^3 = (V_8^2 \oplus K_{14}^3) \oplus V_{14}^3 \oplus V_{16}^3$$

So, $T_1 \oplus T_2 \oplus T_3 \oplus T_4$ is precisely

$$P_5 \oplus P_7 \oplus P_8 \oplus V_6^3 \oplus V_8^3 \oplus V_{14}^3 \oplus V_{16}^3 \oplus K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3$$

which has bias $-1/32$.

§4.3 A Linear Attack on an SPN

Next step: Replace the terms V_i^3 in the above formula by expressions involving U_i^4 and further key bits as follows: notice that
$V_6^3 \oplus K_6^4 = U_6^4 \implies V_6^3 \oplus K_6^4 \oplus K_6^4 = U_6^4 \oplus K_6^4 \implies V_6^3 = U_6^4 \oplus K_6^4$

§4.3 A Linear Attack on an SPN

Next step: Replace the terms V_i^3 in the above formula by expressions involving U_i^4 and further key bits as follows: notice that

$V_6^3 \oplus K_6^4 = U_6^4 \implies V_6^3 \oplus K_6^4 \oplus K_6^4 = U_6^4 \oplus K_6^4 \implies V_6^3 = U_6^4 \oplus K_6^4$

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

§4.3 A Linear Attack on an SPN

Next step: Replace the terms V_i^3 in the above formula by expressions involving U_i^4 and further key bits as follows: notice that
$V_6^3 \oplus K_6^4 = U_6^4 \implies V_6^3 \oplus K_6^4 \oplus K_6^4 = U_6^4 \oplus K_6^4 \implies V_6^3 = U_6^4 \oplus K_6^4$

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

So we get the random variable

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

where

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

§4.3 A Linear Attack on an SPN

Next step: Replace the terms V_i^3 in the above formula by expressions involving U_i^4 and further key bits as follows: notice that
$V_6^3 \oplus K_6^4 = U_6^4 \implies V_6^3 \oplus K_6^4 \oplus K_6^4 = U_6^4 \oplus K_6^4 \implies V_6^3 = U_6^4 \oplus K_6^4$

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

So we get the random variable

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

where

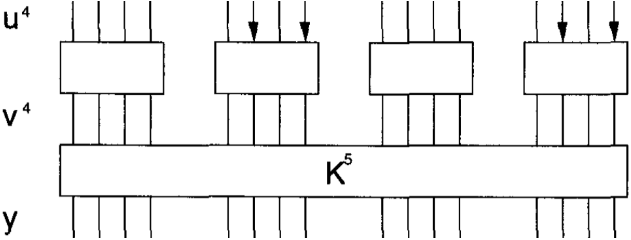
$$\overline{K} = K_5^1 \oplus K_7^1 \oplus K_8^1 \oplus K_6^2 \oplus K_6^3 \oplus K_{14}^3 \oplus K_6^4 \oplus K_{14}^4 \oplus K_8^4 \oplus K_{16}^4.$$

Since the key is fixed, $\overline{K}$ has the (fixed) value 0 or 1. It follows that

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

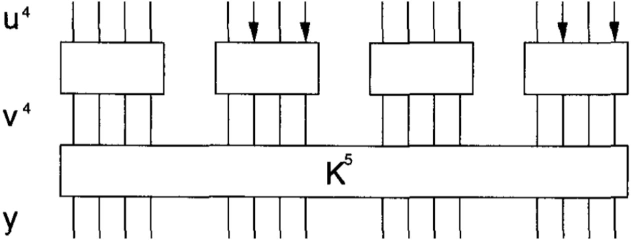
has bias equal to $\pm 1/32$, where the sign of the bias depends on the values of the unknown key bits.

§4.3 Extracting Key Bits



Our linear expression $P_5 \oplus P_7 \oplus P_8 \oplus U^4_6 \oplus U^4_8 \oplus U^4_{14} \oplus U^4_{16} = 0$ involves four input bits of U^4.

§4.3 Extracting Key Bits

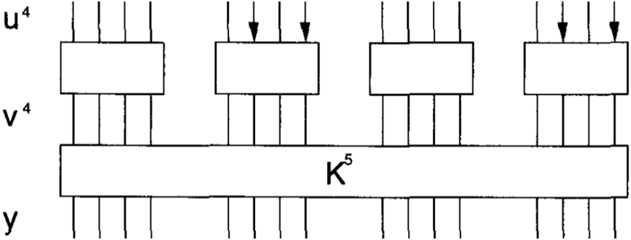


Our linear expression $P_5 \oplus P_7 \oplus P_8 \oplus U^4_6 \oplus U^4_8 \oplus U^4_{14} \oplus U^4_{16} = 0$ involves four input bits of U^4. If we have a plaintext-ciphertext pair, there are 8 key bits, namely

$$K^5_5, K^5_6, K^5_7, K^5_8, K^5_{13}, K^5_{14}, K^5_{15}, K^5_{16}$$

needed to partially decrypt back to the level of U^4.

§4.3 Extracting Key Bits



Our linear expression $P_5 \oplus P_7 \oplus P_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4 = 0$ involves four input bits of U^4. If we have a plaintext-ciphertext pair, there are 8 key bits, namely

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

needed to partially decrypt back to the level of U^4. We don't need to guess the entire key for the last round! Thus we only need to guess $2^8 = 256$ values, instead $2^{16} = 65546$, which is huge difference.

§4.3 Testing Key Guesses

- Let $\mathcal{J}$ be a set of plaintext-ciphertext pairs (x, y). $|\mathcal{J}| = \tau$.

§4.3 Testing Key Guesses

- Let $\mathcal{J}$ be a set of plaintext-ciphertext pairs (x, y). $|\mathcal{J}| = \tau$.
- Start with the guess of $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle}) = (0000, 0000)$ for relevant key bits.

§4.3 Testing Key Guesses

- Let $\mathcal{J}$ be a set of plaintext-ciphertext pairs (x, y). $|\mathcal{J}| = \tau$.
- Start with the guess of $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle}) = (0000, 0000)$ for relevant key bits.
- For each 256 possible guesses $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle})$ do the following:
    - Initialize the counter to zero
    - For each $(x, y) \in \mathcal{J}$, do a partial decryption to get $u^4_{\langle 2\rangle}$ and $u^4_{\langle 4\rangle}$.
    - Compute the value of $p_5 \oplus p_7 \oplus p_8 \oplus u^4_6 \oplus u^4_8 \oplus u^4_{14} \oplus u^4_{16}$.
    - Increment the counter if this sum is zero.

§4.3 Testing Key Guesses

- Let $\mathcal{J}$ be a set of plaintext-ciphertext pairs (x, y). $|\mathcal{J}| = \tau$.
- Start with the guess of $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle}) = (0000, 0000)$ for relevant key bits.
- For each 256 possible guesses $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle})$ do the following:
    - Initialize the counter to zero
    - For each $(x, y) \in \mathcal{J}$, do a partial decryption to get $u^4_{\langle 2\rangle}$ and $u^4_{\langle 4\rangle}$.
    - Compute the value of $p_5 \oplus p_7 \oplus p_8 \oplus u^4_6 \oplus u^4_8 \oplus u^4_{14} \oplus u^4_{16}$.
    - Increment the counter if this sum is zero.
- At the end of this counting process, we expect that most counters will have value close to $\tau/2$, but the counter with the correct candidate $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle})$ will have a value close to $\tau/2 \pm \tau/32$.

§4.3 A Linear Attack on an SPN

Example: Consider our key segment guess of $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle}) = (0000, 0001)$.

§4.3 A Linear Attack on an SPN

Example: Consider our key segment guess of $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle}) = (0000, 0001)$.

Suppose that we have a plaintext-ciphertext pair $(1820, 8BF4)$. Hence

$$P : 0001\,1000\,0010\,0000$$

$$C : 1000\,1011\,1111\,0100$$

§4.3 A Linear Attack on an SPN

Example: Consider our key segment guess of $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle}) = (0000, 0001)$.

Suppose that we have a plaintext-ciphertext pair $(1820, 8BF4)$. Hence

$$P : 0001\,1000\,0010\,0000$$

$$C : 1000\,1011\,1111\,0100$$

Therefore output of S-box S^4_2 is: $(1011) \oplus (0000) = (1011) = B$, and the output of S^4_4 is $(0100) \oplus (0001) = (0101) = 5$.

§4.3 A Linear Attack on an SPN

Example: Consider our key segment guess of $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle}) = (0000, 0001)$.

Suppose that we have a plaintext-ciphertext pair $(1820, 8BF4)$. Hence

$$P : 0001\,1000\,0010\,0000$$

$$C : 1000\,1011\,1111\,0100$$

Therefore output of S-box S^4_2 is: $(1011) \oplus (0000) = (1011) = B$, and the output of S^4_4 is $(0100) \oplus (0001) = (0101) = 5$.

Thus inputs are $\pi_S^{-1}(B) = 6 = (0110)$ and $\pi_S^{-1}(5) = C = (1100)$.

§4.3 A Linear Attack on an SPN

Example: Consider our key segment guess of $(K^5_{\langle 2\rangle}, K^5_{\langle 4\rangle}) = (0000, 0001)$.

Suppose that we have a plaintext-ciphertext pair $(1820, 8BF4)$. Hence

$$P : 0001\,1000\,0010\,0000$$

$$C : 1000\,1011\,1111\,0100$$

Therefore output of S-box S^4_2 is: $(1011) \oplus (0000) = (1011) = B$, and the output of S^4_4 is $(0100) \oplus (0001) = (0101) = 5$.

Thus inputs are $\pi_S^{-1}(B) = 6 = (0110)$ and $\pi_S^{-1}(5) = C = (1100)$.

So we get

$$p_5 \oplus p_7 \oplus p_8 \oplus u^4_6 \oplus u^4_8 \oplus u^4_{14} \oplus u^4_{16} = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

Do not increment the counter. Hold the key fixed, and move on to next pair in the set $\mathcal{J}$.

§4.3 A Linear Attack on an SPN

How many pairs do we need to check?

§4.3 A Linear Attack on an SPN

How many pairs do we need to check?

Experimental evidence indicates that the number of plaintext-ciphertext pairs needed for a successful linear attack is proportional to $1/\epsilon^2$, that is, $\tau \approx c/\epsilon^2$ for some small constant c.

§4.3 A Linear Attack on an SPN

How many pairs do we need to check?

Experimental evidence indicates that the number of plaintext-ciphertext pairs needed for a successful linear attack is proportional to $1/\epsilon^2$, that is, $\tau \approx c/\epsilon^2$ for some small constant c.

In our case, where $c = 8$ and $\epsilon = 1/32$, $\tau \approx 8 \times 32^2 \approx 8000$.

§4.3 A Linear Attack on an SPN

How many pairs do we need to check?

Experimental evidence indicates that the number of plaintext-ciphertext pairs needed for a successful linear attack is proportional to $1/\epsilon^2$, that is, $\tau \approx c/\epsilon^2$ for some small constant c.

In our case, where $c = 8$ and $\epsilon = 1/32$, $\tau \approx 8 \times 32^2 \approx 8000$.

How many partial decryptions for $\tau \approx 8000$?

§4.3 A Linear Attack on an SPN

How many pairs do we need to check?

Experimental evidence indicates that the number of plaintext-ciphertext pairs needed for a successful linear attack is proportional to $1/\epsilon^2$, that is, $\tau \approx c/\epsilon^2$ for some small constant c.

In our case, where $c = 8$ and $\epsilon = 1/32$, $\tau \approx 8 \times 32^2 \approx 8000$.

How many partial decryptions for $\tau \approx 8000$? We have 256 guesses for the relevant key segment and so there are approximately 16×10^6 partial decryptions. Compare it with $2^{32} \approx 4.3 \times 10^9$.

§4.3 A Linear Attack on an SPN

How many pairs do we need to check?

Experimental evidence indicates that the number of plaintext-ciphertext pairs needed for a successful linear attack is proportional to $1/\epsilon^2$, that is, $\tau \approx c/\epsilon^2$ for some small constant c.

In our case, where $c = 8$ and $\epsilon = 1/32$, $\tau \approx 8 \times 32^2 \approx 8000$.

How many partial decryptions for $\tau \approx 8000$? We have 256 guesses for the relevant key segment and so there are approximately 16×10^6 partial decryptions. Compare it with $2^{32} \approx 4.3 \times 10^9$.

How about brute force attack for a realistic size key?

- Brute force on 128 bit key $\approx 3.4 \times 10^{38}$ keys.
- Test one trillion keys per sec: 10^{19} years to check all.

§4.3 A Linear Attack on an SPN

```
Linear Attack $(\tau, T, \pi_S^{-1})$:

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$[L_1, L_2] + 1$
```

§4.3 A Linear Attack on an SPN

Linear Attack (τ, T, π_S^{-1}) **continues:**

```
max ← $-1$
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 ← Count$[L_1, L_2]$
        maxkey ← $(L_1, L_2)$
output (maxkey)
```