Cryptography and Network Security
Scribe Week 7 Class 3

Pravesh Jain

October 2020

Introduction

In the following lecture, we will discuss more about Linear Cryptanalysis and how to attack a GPig Cipher using Linear Approximation, and a linear approximation table.

S-Box

S-Box is a non linear layer that takes , say m bit inputs and convert it into a n bit output.
Input Tuple: $(x_1, x_2, ..., x_n)$ where $x_i \epsilon \{0, 1\}$
Output Tuple: $y_1, y_2, ..., y_m$ where $y_i \epsilon \{0, 1\}$
Output are not independent among themselves or from the inputs.

Computing the probability of linear approximation

Let's say there's an output vector $Alpha = \{y_1, y_2, ..., y_m\}$, such that for no input vector S$(x_1, x_2, ..., x_n) \neq Y_1$ then
Pr$(X_1 = x_1, X_2 = x_2, ..., X_n = x_n, Y_1 = y_1, Y_2 = y_2..., Y_m = Y_n) = 0$
If it is a function ,i.e, S$(x_1, x_2, ..., x_n) = Alpha$ then,
Pr$(X_1 = x_1, X_2 = x_2, ..., X_n = x_n, Y_1 = y_1, Y_2 = y_2..., Y_m = Y_n) = 2^{-m}$

As a attacker we will choose the combination of variable such that we obtain a higher absolute value of bias.

Representing the Approximation

In order to represent the approximation so that the attacker can decide which to choose, he uses a tabular form to store the data. The table is known as Linear Approximation Table. We follow the following steps to create the table:

- Any linear approximation of will look like as follows

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

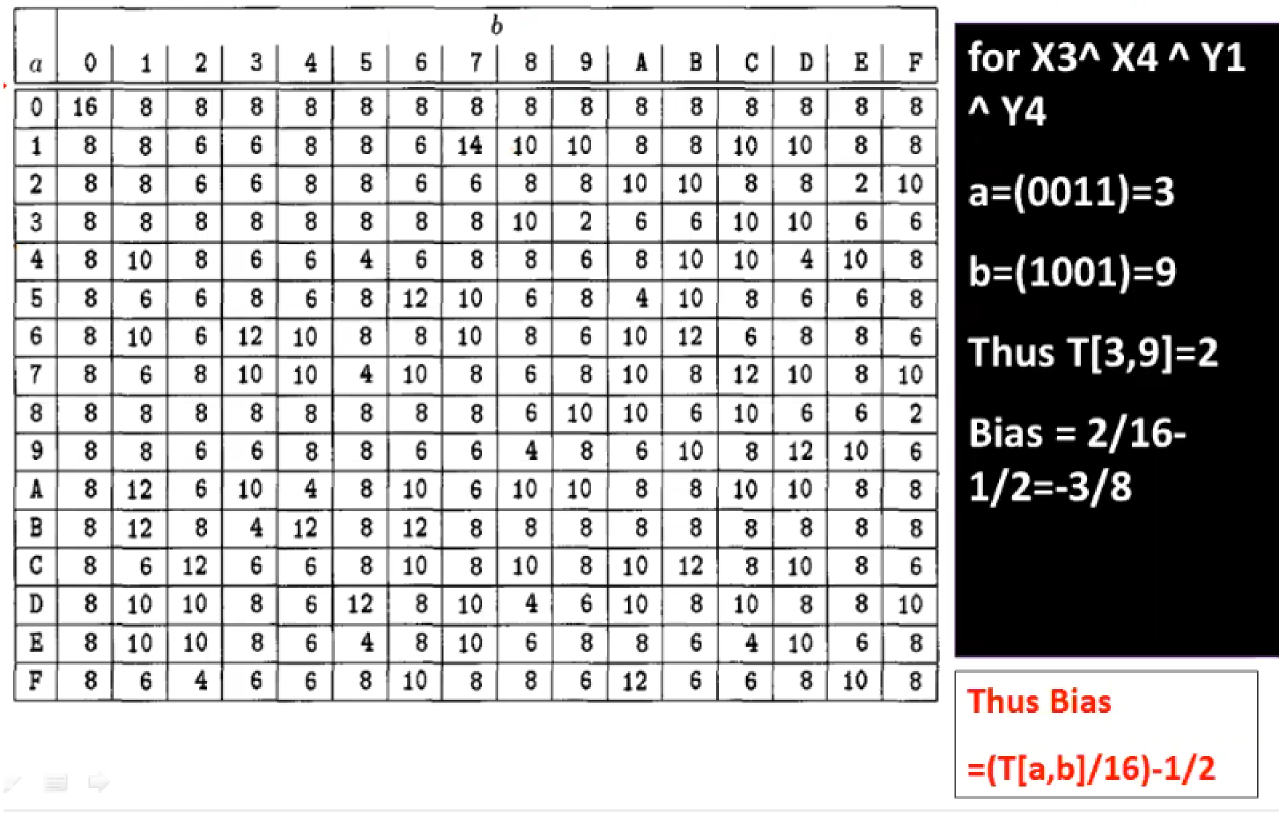
  where $a_i \epsilon \{0, 1\}$ and $b_i \epsilon \{0, 1\}$

- Let's say a = {1, 0, 1, 1} then a can be represented by Hexadecimal number between to 0 to F. In this case, a = 11. These number can now be stored in a table.

- Let's say $\vec{a} = \{a_1, a_2, ..., a_n\}$ and $\vec{b} = \{b_1, b_2, ..., b_m\}$ then the above equation can be written as $\vec{a} \, . \, \vec{x} \oplus \vec{b} \, . \, \vec{y} = 0$. This is the linear notation of representation.

- In the each individual cell of the table we store the bias corresponding to that table.

The sample table will look like



	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

for X3^ X4 ^ Y1 ^ Y4

a=(0011)=3

b=(1001)=9

Thus T[3,9]=2

Bias = 2/16-1/2=-3/8

Thus Bias =(T[a,b]/16)-1/2

As you can see, the row are indexed using A, and columns are indexed using B

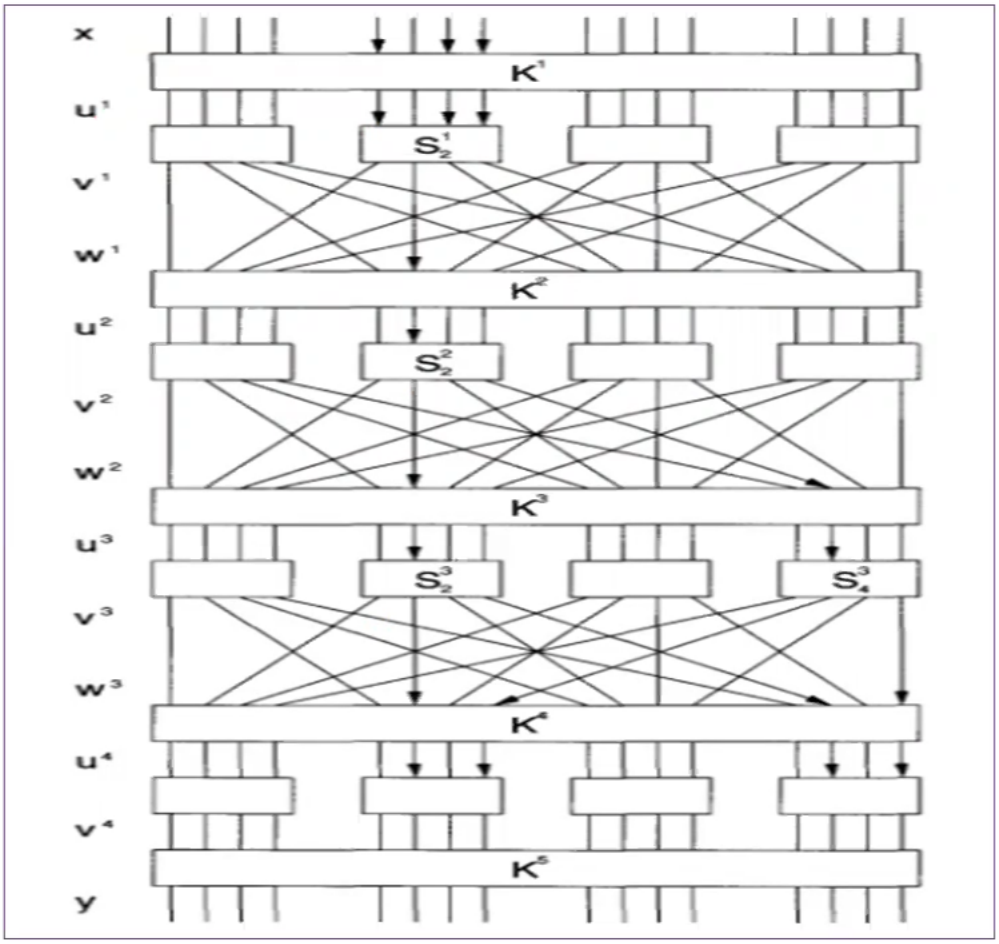
Linear Attack

For a good attack we need to satisfy the following conditions:

- We need to form a linear approximation, involving the plain-text, key and the state before the last rounds, which has a good bias (generally means non zero bias)
- The non-linear components in the ciphers are only the S-Boxes.
- So, we use Linear Approximation Table (formed in the last section) to obtain a good linear approximation (high absolute value of bias)

Now we will see how to crack the cipher using our Linear Approximation Table

We need to crack the following cipher.



As we can see there are 4 S-Box in the first layers, whose input are u^1 and output are v^1. These outputs are permuted and are now denoted with w^1. This notation is carried out further. Each denoting rounds of the cipher.

The arrows indicate the bits that we will consider for our analysis. S_2^1

Next we will calculate bias for each of the S-Boxes.

- 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 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 $-1/4$

All the bias are corresponding the table listed about.

We assume that the four random variable are Independent, and hence we can pile them up using the Piling Up Lemma.

To find out the approximation of our cipher, we will individually find out approximation of the S-boxes. As we did above. We first find the approximation for S_2^1 then, from the bit that are disturbed. Then we will carry out similarly approximation for the bits that are disturbed by our S-Box, and use them to approximate the subsequent S-Boxes.
After approximating S_2^2 we find out that the two subsequent bit's that being affected by output of S_2^2 are S_2^3 and S_4^3 . And now we approximate S_2^3 and S_4^3 . Now we have figured out all the bias, we will use Piling Up Lemma.

Our final bias for $T_1 \oplus T_2 \oplus T_3 \oplus T_4 = 0$ will be $2^3 * (-1/4)^3 * (1/4) = -1/32$

For the next step, we will substitute U_5^1 in terms of $X_5 \oplus K_5^1$, and carry a similar substitution for all the terms involved in the expression.

Finally, all the same terms will be cancelled as they are in $\oplus$ with each other. Our end goal is to express our final expression in terms of last layer input bit, key and plain text bit. This is helpful because finally we are focused on the bias only, and key bits will have a constant contribution to the bias, whatever may be their value. Hence, we can safely drop them for our evaluation.

The final expression that remains after applying all the substitution and ignoring the key values is $X_5 \oplus X_7 \oplus X_8 \oplus U_6^4 \oplus U_8^4 \oplus U_{14}^4 \oplus U_{16}^4$. This expression can either be 1 or 0. Now we have reduced all the middle layers of ciphers, and are at the final step of decryption, effectively reducing the complexity of the cipher.

Now as we have reduced our final output to 2 S-Boxes in final layer, we don't need to completely guess the keys, we will guess the those S-Box alone. Effectively reducing the space for brute forcing the Key 5. We need to guess only parts of K_5 and only 2^8 possible cases.

We can summarise the attack as follows

- The expression has U^4 where are in the second and fourth S-Box of the last round
- The attacker has cipher text corresponding to a large number of plain text
- He guesses 8 Key bits $K_5[5-8], K_5[13-16]$
- Make a frequency table, where for each key a count is stored to denote the number of cases the above expression satisfies
- If we inspect T plain-text, cipher-text pairs, then for a wrong guess T/2 cases satisfy.
- For right guess, T/2 $\overset{+}{-}$ T/32 cases , the expression is satisfied. 1/32 is due to the bias, which will help us distinguish between wrong and right guess.
-

Conclusion

In this lecture we saw how to use bias, and Linear Approximation Table to guess the key for GPig cipher. We figured out how to divide the key into smaller parts, so as to make smaller number of calculation. Effectively reducing the amount of calculation required to guess the key. Also, for guessing the correct keys, we learnt how to find out the bias so that we can figure out which guess to ignore and which to use for our purpose.