LINEAR CRYPTANALYSIS

210050118 - PLS LOKESH REDDY
210050119 - P HARI PRAKASH REDDY

May 1, 2023

1 Introduction

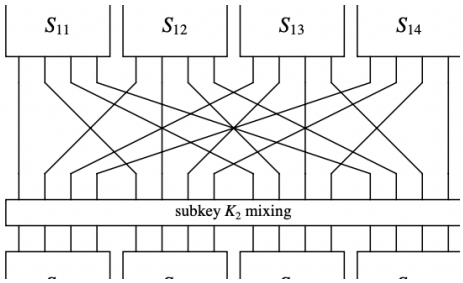
- Linear cryptanalysis is a general type of cryptanalysis based on discovering affine approximations to a cipher's action in cryptography.
- In this project, we implemented a ***Linear Cryptanalysis*** attack on a basic **SPN(Substitution-Permutation Network)** cipher
- Here we implemented an attack on SPN cipher with **3 rounds and a 48bit key**
- The 3 main components of an SPN cipher are
  - Substitution
  - Permutation
  - Key Mixing

1.1 Substituion

- The 16-bit plaintext is divided into 4 parts, where each part is passed through an S-box and gets a 4-bit output.
- The most fundamental property of an S-box is that it is a nonlinear mapping, i.e., the output bits cannot be represented as a linear operation on the input bits.
- For the sake of simplicity, in our cipher we chose the 4 S-boxes in a round as the same

1.2 Permutation

- The permutation portion of a round is simply the transposition of the bits or the permutation of the bit positions
- The output i of S-box j is connected to input j of S-box i



1.3 Key Mixing

- To achieve the key mixing, we use a simple bit-wise exclusive-OR between the key bits associated with a round (referred to as a subkey) and the data block input to a round

1.4 Decryption

- In order to decrypt, data is essentially passed backwards through the network
- However, the mappings used in the S-boxes of the decryption network are the inverse of the mappings in the encryption network
- This implies that in order for an SPN to allow for decryption, all S-boxes must be bijective

2 Cryptanalysis

The sbox which we want to use can be written in a file **sbox.txt**. The code will generate the Linear Approximation table and uses the linear equations with high bias

2.1 Extracting Round3 Key

- Say our plaintext is P1P2P3..P16, and ciphertext is C1C2C3...C16 .In each round we do key mixing with a 16bit key
- For an S-box with input X1X2X3X4 and output Y1Y2Y3Y4, the probability that $X1 \oplus X3 \oplus X4 = Y2$ is 3/4, i.e a bias of +1/4

- Let Ui be input to S-boxes on round i and Vi be the output of S-boxes
- Now $P_5 \oplus K_{1,5} = U_{1,5}, P_7 \oplus K_{1,7} = U_{1,7}, P_8 \oplus K_{1,8} = U_{1,8}, V_{1,6} \oplus K_{2,6} = U_{2,6}$
- The probability that $U_{1,5} \oplus U_{1,7} \oplus U_{1,8} = V_{1,6}$, i.e $P_5 \oplus P_7 \oplus P_8 \oplus U_{2,6} \oplus \sum K = 0$ is 3/4. $\sum K = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6}$
- That means $P_5 \oplus P_7 \oplus P_8 \oplus U_{2,6} = 0$ have a probability of 1/4 or 3/4 depending on $\sum K$. However the abs(bias)=1/4.
- So in our attack for each plaintext/ciphertext pair we choose K3[4:8] from 0 to 15 and find U2,6 and check the above relation. If it becomes true we increment the score of that 4bit key.
- After enumerating through all plaintext/ciphertext pairs we find the biases for each 4bit key and the one with the highest bias is our 4bit key. This is in fact found to be the partial subkey of K3.

2.2 Extracting Round2 key

- Now that we have Round3 Key, we can find the output of KeyMixing with Round2 Key, by going through network in backward manner
- Now this is simply an attack on 2 round cipher
- We follow a similar procedure as above, however now we choose a 256bit partial subkey.
- Say we choose $[K_{2,2}K_{2,4}K_{2,6}K_{2,8}K_{2,10}K_{2,12}K_{2,14}K_{2,16}]$ as our target partial subkey, we iterate over all possible values of target subkey i.e from 0 to 255
- For each key we would check for all plaintext/ciphertext pairs the relation below
- $P_5 \oplus P_6 \oplus P_7 \oplus P_8 \oplus P_{13} \oplus P_{14} \oplus P_{15} \oplus P_{16} \oplus U_{1,5} \oplus U_{1,6} \oplus U_{1,7} \oplus U_{1,8} \oplus U_{1,13} \oplus U_{1,14} \oplus U_{1,15} \oplus U_{1,16} = 0$
- The U1,i represents the calculated Pi from our guess of target partial subkey
- If the above relation holds true we increment the count
- We now take the key with highest absolute bias, where bias=abs(count/5000 - 0.5)
- The other part of round2 key can be found in a similar way as above

2.3 Extracting Round1 key

- Having round2 key, we can find the output of input $\oplus$ round1 by going through network backwards
- We can find the round1 key by xoring the input with above output

3 Verfication

- The attacker can get a bunch of possible keys from the cryptanalysis
- He could verify which of them is true by iterating over each possible key and for each key, checking each plaintext/ciphertext pair with the possible key
- The correct key should give same results as ciphertexts on encrypting the plaintext

The source code for the above implementation can be found here