CS6160 Assignment-2

Vishal Vijay Devadiga (CS21BTECH11061)

Approach

The paper in the references describes a method called the **single byte bias attack** to find the passcode using the keystream distribution at a particular position and the ciphertexts of the same plaintext. The method is based on the fact that the keystream distribution at a particular position is biased.

Algorithm

Below is the algorithm used to find the passcode:

Input:-

- Ciphertexts C_i of the same plaintext
- Position of the plaintext r
- Keystream distribution $p_{r,k}$ at position r and key k

Output:-

- The passcode at position r

Algorithm:-

- Let the number of ciphertexts be n
- Let $S_i = 0$ for all $i \in [0, 255]$
- For i in range 0 to n-1:
  - Set $S_{C_i[r]} = S_{C_i[r]} + 1$
- For u in range 0 to 10:
  - Let $\lambda_u = 0$
  - For k in range 0 to 32:
    - $N_k^u = S_{k \oplus u}$
    - $\lambda_u = \lambda_u + N_k^u \cdot \log(p_{r,k})$
- Return the value of u for which λ_u is maximum

Answer

The passcode is `475103`.

Observation

The only variable in this algorithm that we can change is the amount of samples in the probability distribution. The more the samples, the more accurate the result.

Below is the set of answers for different number of samples (2^n) executed 5 times:

n	P1	P2	P3	P4	P5
8	471687	171596	911439	762535	309887
10	076213	079705	970685	272587	571606
12	271094	576815	975397	777128	575380
14	475155	876938	478503	375253	573761
16	472153	872103	472553	475106	575105
18	475103	478103	475103	475103	475105
20	475103	475103	475103	475103	475103
22	475103	475103	475103	475103	475103
24	475103	475103	475103	475103	475103

From the above table, we can see that on increasing the number of samples, the passcode converges to `475103`. On lower number of samples, the passcode is not accurate as the probability distribution is not accurate. With more samples, the probability distribution is more accurate and the passcode is also accurate.

Quirks related to the assignment:

- The value of the passcode at position r is between 48 and 57 rather than 0 and 9. This is probably because the passcode was provided as a string in the plaintext.
- One random observation related to python: `[[0]*256]*6` creates a list of 6 references to the same list `[0]*256` rather than creating 6 different lists. This caused a lot of confusion for me as I was updating the list of lists and all the lists were getting updated. I had to change the initialization to `[[0]*256 for _ in range(6)]` to fix this issue.