REPORT

Challa Akshay Santoshi – CS21BTECH11012

OBSERVATIONS

keystream_probs.txt contains the keystream probability distribution at each position.
xor_distribution.txt contains the distribution at each position for each guess (0-9).

- For the keystreams, most of the probabilities fall within a narrow range between 0.028 and 0.032.
  However, at position 2, '0' has a significantly higher probability of approximately 0.062, which is roughly double the probabilities of other byte values.
- The existence of this high probability for specific byte suggests that there is an underlying bias and that it is not truly random.
  So, this can be exploited and an attack can be done by analysing how these keystream distribution correlates with ciphertexts to get successful guesses of passcode.

APPROACH

- We retrieve the encrypted passcode by analysing the probability distributions of keystream and XOR distributions.
- First, we use Key Scheduling Algorithm to generate pseudo-random keystreams which give distributions at each position. This is used as reference.
- We generate 2^{24} keystream values and estimate the probability distribution for each byte at each position in the passcode.
- This keystream distribution is then compared to ciphertext xor distributions generated from the files.
- We calculated the XOR distribution for each possible digit (0-9) at each position of the passcode by xoring with corresponding ciphertext values.
- We identified digits by matching each XOR distribution at a position to the precomputed keystream distribution.

- When a distribution matches closely (<= 0.0003), the corresponding digit is considered for the passcode.

CRACKED PASSCODE

475103