Scribe: Cryptography and Network Security (Class.5.B)

Juluri Shree Shiva Teja

26-Sep-2020

1 Introduction

This scribe mainly introduces the estimation of attack on cryptographic algorithms with known ciphertext assuming that the attacker has infinite computation power.

2 Unicity distance

The least amount of plaintext which can be deciphered uniquely from the corresponding ciphertext given unbounded resources by the attacker.
Cryptographic algorithms whose ciphers are used within unicity cannot be broken even by exhaustively trying all the possible keys. These ciphers are hence unbreakable even with brute force attack with infinite computation power

3 Entropy of Plain Text

If every English letter is equally probable, then
entropy H = $\log_2 26 = 4.76$
But English language has standard frequency distribution among the alphabet,then
entropy H = 4.19
It clearly indicates that randomness is reduced when we consider probability distribution. It can be further reduced with analysis taking into account of bigrams, trigrams etc.

3.1 Redundancy

P_n be the random variable which denotes the probability distribution of n-grams of plaintext and hence $\log_2 \mid P \mid$ gives the entropy of random language
$H_L = \lim_{n\to\infty} \frac{H(P_n)}{n}$ denotes the entropy of natural language L

Redundancy of language L = $R_L = 1 - \frac{H_L}{\log_2 |P|}$
Redundancy gives the fraction of excess letters i.e measure of length which is more than optimal length. Its value is always between 0 and 1 as entropy of language is maximum when it is purely random which indicates every letter is equally probable. It indicates the percentage of optimal length (obtained with encoding) when compared to normal language. In case of proper encoding, the redundancy will be zero.

4 Spurious Keys

If you know a plaintext is taken from a 'natural' language then knowing the ciphertext rules out a certain subset of the keys. Of the remaining possible keys only one is correct. The remaining possible, but incorrect, keys are called the spurious keys.

4.1 Key equivocation

H(K | C) is the amount of randomness that remains of the key given the cipher text. Cryptographic algorithm should be designed in order to maximize it. It is an important measure to determine the unicity distance.

4.2 Lower Bound of equivocation of key

Let P^n denotes random variable representing n-gram plaintext and C^n denotes random variable representing n-gram ciphertext.
Assuming n to be large, we have
H(P^n) $\approx nH_L = n(1 - R_L) \log_2 | P |$ from the above equations for entropy
H(C^n) $\leq n \log_2 | C |$ as RHS denotes the entropy of completely random ciphertext
We know that H(K| C^n) $= H(K) + H(P^n) - H(C^n)$
$\Rightarrow H(K \mid C^n) \geq H(K) + n(1 - R_L) \log_2 | P | - n \log_2 | C |$
$If \mid P \mid = \mid C \mid$, then we have
$\Rightarrow H(K \mid C^n) \geq H(K) - nR_L \log_2 | P |$

4.3 Possible Keys

Let K(y) is the set of all possible keys for which y is the ciphertext for meaningful plaintexts. Only one of them will be denoting the actual key.
Hence, number of spurious keys = $| K(y) | -1$
Let Average number of spurious keys be denoted by

$$s_n = \sum_{y \in C^n} p(y)(| K(y) | -1)$$

$$\Rightarrow s_n = (\sum_{y \in C^n} p(y) \mid K(y) \mid) - 1$$

$$\Rightarrow s_n + 1 = (\sum_{y \in C^n} p(y) \mid K(y) \mid)$$

4.4 Upper bound of equivocation of key

We know from the definition of conditional probability that

$$H(K \mid C^n) = \sum_{y \in C^n} p(y) H(K \mid y)$$

As we defined earlier, K(y) denotes the set of possible keys K given y. Hence,

$$\Rightarrow H(K \mid C^n) = \sum_{y \in C^n} p(y) H(K(y))$$

Since the maximum value of entropy H(X) = $\log_2 \mid X \mid$, we have

$$\Rightarrow H(K \mid C^n) \leq \sum_{y \in C^n} p(y) \log_2 \mid K(y) \mid$$

From the Jensen's inequality, for a real valued strictly concave function, we have for a set of points, the y corresponding to centroid point of them will be lower than y corresponding to x-coordinate of centroid lying on the curve. Hence, we have

$$\Rightarrow H(K \mid C^n) \leq \log_2 (\sum_{y \in C^n} p(y) \mid K(y) \mid)$$

As proved earlier, we can have

$$\Rightarrow H(K \mid C^n) \leq \log_2 (s_n + 1)$$

5 Conclusion

From the above upper bounds and lower bounds of H(K$\mid C^n$), we can write

$$H(K) - nR_L \log_2 \mid P \mid \leq H(K \mid C^n) \leq \log_2 (s_n + 1)$$

If the keys are equally probable, then H(k) would be at its maximum value of $\log_2 \mid K \mid$. In that case, we have

$$\log_2 (s_n + 1) \geq \log_2 \mid K \mid - nR_L \log_2 \mid P \mid$$

$$\Rightarrow \log_2 (s_n + 1) \geq \log_2 \mid K \mid - \log_2 \mid P \mid^{nR_L}$$

$$\Rightarrow s_n + 1 \geq \frac{\mid K \mid}{\mid P \mid^{nR_L}}$$

$$\Rightarrow s_n \geq \frac{\mid K \mid}{\mid P \mid^{nR_L}} - 1$$

As we want average number of spurious keys s_n to be zero, equating the lower bound of it to zero gives

$$n_0 = \frac{\log_2 \mid K \mid}{R_L \log_2 \mid P \mid}$$

This n_0 gives the value of unicity distance which is nothing but minimum number of ciphertexts required to make number of spurious keys to zero. The accuracy increases with increasing n as we have made assumption that n is sufficiently large while proving this. Also, this result holds when keys are equally likely. Hence, if we have a ciphertext of length n_0, we might be able to extract the key even if user has infinite computation power. Hence, we can conclude that key size alone doesnt define the strength of algorithm.