Scribe: Cryptography and Network Security (Week 4, Class 3)

Chandan Kumar

28-Sep-2020

1 One Time Pad

An encryption technique in which each letter of the plain text are given some specific binary digits of fixed length, and keys are streams of binary number randomly generated. A key (pad) once used will not be used again and hence each time a plaintext is encrypted, it is encrypted with random key (pad). Ciphertext are generated using XOR operation of plaintext and key. Let us take an example to understand it in a better way with encodings for characters as $\{e = 000, h = 001, i = 010, k = 011, l = 100, r = 101, s = 110, t = 111\}$

$$plaintext \oplus key = ciphertext \tag{1}$$

Actual text	h	e	i	l	h	i	t	l	e	r
plaintext	001	000	010	100	001	010	111	100	000	101
key	111	101	110	101	111	100	000	101	110	000
ciphertext	110	101	100	001	110	110	111	001	110	101

Table 1: OTP example

OTP is said to be unconditionally secure because for a ciphertext of same size as the plaintext, there is a equi-probable key that produces it. It can be shown in the same way a shift cipher has been proved and hence along with shift cipher OTP is also an example of perfect cipher. Some of the practical problems of OTP can be listed as follow,

1. Large quantities of random keys are necessary.
2. Increases the problem of key distribution.
3. Thus we will continue to search for ciphers where one key can be used to encrypt a large string of data and still provide computational security. Like DES (Data Encryption Standard)

OTP can be summarise as follows,

1. Cipher-text provides no information about plaintext
2. All plaintexts are equally likely.
3. Pad must be random, used only once
4. Pad is known only by sender and receiver
5. Pad is same size as message
6. No assurance of message integrity

2 Entropy

Entropy is the measure of randomness of bits/texts of any plain-text, ciphertext, key-space etc. Let X be a discrete random variable which takes on values from a finite set. Then the entropy of random variable X can be given by,

$$H(X) = -\sum x \in XPr[x]log_2Pr[x] \tag{2}$$

Let us understand the entropy and it's use with the help of an example. Following are the given details in order to calculate the entropy of plaintext i.e H(P).
$P = \{a, b\}, P_p(a) = 1/4, P_p(b) = 3/4$
$K = \{k_1, k_2, k_3\}, P_k(k_1) = 1/2, P_k(k_2) = P_k(k_2) = 1/4$
$H(P) = 1/4log_2(4) + 3/4log_2(4/3) = 0.81$

2.1 Huffman Encoding

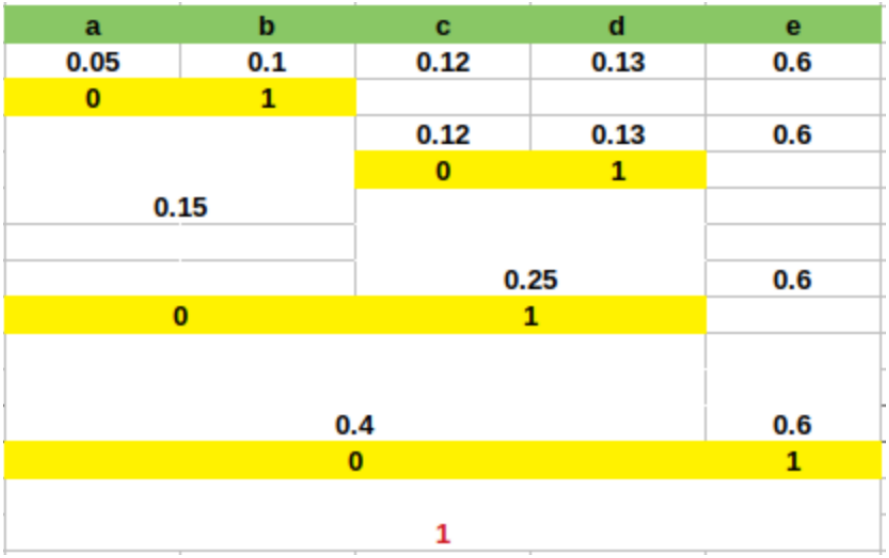
Huffman encoding is one example which encodes message such that the average length of ciphertext is as short as H(P). Let us is consider the messages from X: x1,x2,. . . ,xk. We need to see the value of H(X) if it is equal to average length or not.

Steps covered for Huffman encoding can be informally summarised as,

1. The message set X has a probability distribution. It should be arranged in ascending order.
2. Initially the codes of each element are empty.
3. Choose the two elements with minimum probabilities.
4. Merge them into a new letter, say x12 with probability as the sum of x1 and x2. Encode the smaller letter 0 and the larger 1.
5. When only one element remains, the code of each letter can be constructed by reading the sequence backwards.

2.2 Example Illustrating Huffman coding

Let us take an example to understand the technique involved in Huffman coding.
$X = \{a, b, c, d, e\}$
$P(a) = .05, P(b) = .10, P(c) = .12, P(d) = .13, P(e) = .6$
$H(x) = 1.7402$



So the codes obtained for each alphabet can be written as, $\{a = 000, b = 001, c = 010, d = 011, e = 1\}$
Now applying the average rule on each alphabet to calculate the l(F),

$$l(f) = \Sigma_{x \in X} Pr[x]|f(x)| \tag{3}$$

Putting values in the above equation, we have,
$l(f) = .05 * 3 + .10 * 3 + .12 * 3 + .13 * 3 + .6 * 1 = 1.8$
It can be seen that l(f) is approximately equal to H(X).It is proved at the end that $H(X) <= l(f) <= H(X) + 1$

2.3 Some more results on entropy

Let X and Y are random variables.

1. $H(X, Y)H(X) + H(Y)$, When X and Y are independent:
2. $H(X, Y) = H(X) + H(Y)$,
3. $H(X|Y) = -p(x|y)log2p(x|y)$
4. $H(X, Y) = H(Y) + H(X|Y)$
5. $H(X|Y)H(X)$
6. When X and Y are independent: $H(X|Y) = H(X)$

2.3.1 Conditional Entropy

Conditional Entropy is similar to condition probability, it keep relevance to probability in the following way

$$\begin{aligned} H(X|Y=y) &= -\Sigma_x Pr(X=x|Y=y) log(Pr(X=x|Y=y)) \\ H(P,K) &= H(P) + H(K) \\ H(C,K) &= H(C) + H(K|C) \\ H(X|Y) &= \Sigma_y Pr(Y=y) H(X|Y=y) \\ &= \Sigma_x \Sigma_y Pr(Y=y) Pr(X=x|Y=y) * log(Pr(X=x|Y=y)) \end{aligned}$$

Theorem : Let (P,C,K,D,E) be an encryption algorithm then,

$$H(K|C) = H(K) + H(P) - H(C) \tag{4}$$

Proof: Let us first prove that H(P,K)=H(C,K),

$$\begin{aligned} H(P,K,C) &= H(P,K) + H(C|P,K) \\ &= H(C,K) + H(P|C,K) \end{aligned}$$

Hence, H(P,K)=H(C,K), Now from the results from conditional entropy,

$$\begin{aligned} H(C,K) &= H(P,K) \\ H(K|C) + H(C) &= H(P) + H(K) \\ H(K|C) &= H(P) + H(K) - H(C) \end{aligned}$$

The term H(K—C) is also called as equivocation (ambiguity) of Key given in the ciphertext.

3 Perfect vs Ideal Ciphers

If H(P)=H(C), then we have $H(K\|C) = H(K). That is the uncertainty of the key given the cryptogram is the same a$
$H(P\|C) or, equivalently H(C) = H(C\|P)$
$For perfect ciphers, the key size is infinite if the message size is infinite. however if a shorter key size is used then t$

4 Conclusion

The class covered OTP, Entropy and it's implications, Huffman encoding and example, and difference between Ideal and perfect Ciphers.

References

Class lectures