Shannon’s Theory

Debdeep Mukhopadhyay
IIT Kharagpur

Objectives

- Understand the definition of Perfect Secrecy
- Prove that a given crypto-sytem is perfectly secured
- One Time Pad
- Entropy and its computation
- Ideal Ciphers
- Equivocation of Keys

Unconditional Security

- Concerns the security of cryptosystems when the adversary has unbounded computational power, that is has infinite resources.
- Cipher-text only Attack: Attack the cipher using the cipher texts only.
- When is a cipher is unconditionally secured?

A priori and *A posteriori* Probabilities

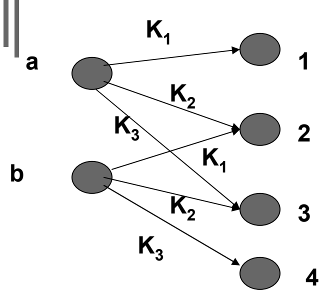
- The plain-text has a probability distribution
- $p_P(x)$: A priori probability of a plain text
- The key also has a probability distribution
- $p_K(K)$: A priori probability of the key.
- The cipher text is generated by applying the encryption function. Thus $y=e_K(x)$ is the cipher text.
- Note, that the plain text and the key are independent distributions.

Attacker wants to compute a posteriori probability of plain text

- The probability distributions on P and K, induce a probability distribution on C, the cipher text.
- For a key K, $C_K(x)=\{e_K(x): x \in P\}$
- Does the cipher text leak information about the plain text?

  Given, the cipher text y, we shall compute the a posteriori probability of the plain text, ie. $p_P(x|y)$ and see whether it matches with that of the a priori probability of the plain text.

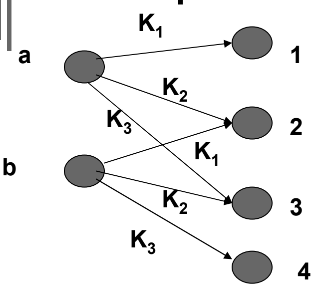
Example



	a	b
K_1	1	2
K_2	2	3
K_3	3	4

- $P=\{a,b\}$; $p_P(a)=1/4$, $p_P(b)=3/4$
- $K=\{K_1,K_2\}$, $p_K(K_1)=1/2$, $p_K(K_2)= p_K(K_3)=1/4$
- $C=\{1,2,3,4\}$. What the a posteriori probabilities of the plain text, given the cipher texts from C?

Example



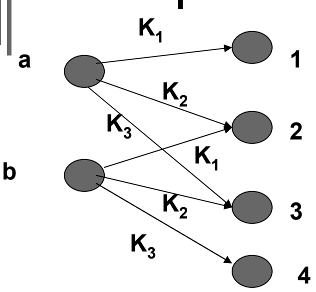
$p_C(1)=p_P(a)p_K(K_1)$
$=(1/4).(1/2)=1/8$

$p_C(3)=p_P(a)p_K(K_3)+p_P(b)$
$p_K(K_2)$
$=(1/4)(1/4)+(3/4)(1/4)=1/16+3/16=1/4$

Likewise I can compute the other probabilities…

$P=\{a,b\}$; $p_P(a)=1/4$,
$p_P(b)=3/4$
$K=\{K_1,K_2\}$, $p_K(K_1)=1/2$,
$p_K(K_2)= p_K(K_3)=1/4$

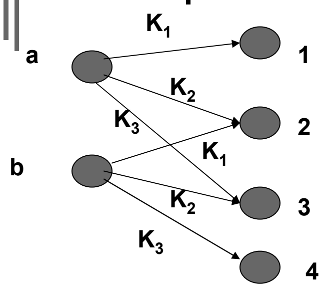
Example



- $p_P(a|1)=1;p_P(b|1)=0$
- $p_P(a|2)=?$
- The '2' can come when the plain text was 'a' and the key was 'K_2' or when the plain text was 'b' and the key was 'K_1'
- Given '2', we need to compute the probability that it came from 'a'.
- Is it that of choosing K_2? No.

$P=\{a,b\}$; $p_P(a)=1/4$,
$p_P(b)=3/4$
$K=\{K_1,K_2\}$, $p_K(K_1)=1/2$,
$p_K(K_2)= p_K(K_3)=1/4$

Example



P={a,b}; $p_P(a)=1/4$, $p_P(b)=3/4$
K={K_1,K_2}, $p_K(K_1)=1/2$, $p_K(K_2)= p_K(K_3)=1/4$

- Given ‘2’, we need to compute the probability that it came from ‘a’.
- The ‘2’ can appear with a probability:
  - by having ‘a’ as the PT and K_2 as the key: (1/4)(1/4)=1/16
  - by having ‘b’ as the PT and K_1 as the key: (3/4)(1/2)=6/16
- $p_P(a|2)=(1/16)/(7/16)=1/7$

Generalization of the Example

$$p_P(x\,|\,y)=\frac{p_P(x)\displaystyle\sum_{K:x=d_K(y)} p_K(K)}{\displaystyle\sum_{\{K:y\in C(K)\}} p_K(K)\,p_P(d_K(y))}$$

Perfect Secrecy

- A Cryptosystem has perfect secrecy if $p_P(x|y)=p_P(x)$ for all x Є P, y Є C.
- That is the a posteriori probability that the plaintext is x, given that the cipher text y is observed, is identical to the a priori probability that the plaintext is x.

Shift Cipher has perfect secrecy

- Suppose the 26 keys in the Shift Cipher are used with equal probability 1/26. Then for any plain text distribution, the Shift Cipher has perfect secrecy.
- Note that $P=K=C=Z_{26}$ and for $0 \le K \le 25$
- Encryption function: $y=e_K(x)=(x+k) \bmod 26$

Perfect Secrecy

$$p_P(x\,|\,y) = \frac{p_P(x)\,p_C(y\,|\,x)}{p_C(y)}$$

$$p_C(y) = \sum_{K \in Z_{26}} p_K(K)\,p_P(d_K(y))$$

$$= \sum_{K \in Z_{26}} \frac{1}{26}\,p_P(y-K) = \frac{1}{26}$$

$$p_C(y\,|\,x) = P_K(y - x \bmod 26)$$

$$= \frac{1}{26}$$

$\textit{Hence}$ Proved

Theorem

- Suppose (P,C,K,E,D) be a cryptosystem, where |K|=|C|=|P|. The cryptosystem offers perfect secrecy if and only if every key is used with probability 1/|K|, and for every xЄP and every y ЄC, there is a unique key, such that $y=e_K(x)$.
  - Perfect Secrecy (equivalent): $p_C(y|x)=p_C(y)$
  - Thus if Perfect Secret, a scheme has to follow the above equation.

Cryptographic Properties

- $p_C(y|x)>0$
- This means that for every cipher text, there is a key, K, st. $y=E_K(x)$
- Thus $|K|\geq|C|$. In our case, $|K|=|C|$
- Thus, there is no cipher text, y, for which there are two keys which take them to the same plaintext.
- There is exactly one key, such that $y=E_K(x)$

One-time Pad

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

Encryption: Plaintext $\oplus$ Key = Ciphertext

	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
	s	r	l	h	s	s	t	h	s	r

One-time Pad

Suppose a wrong key is used to decrypt:

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
"**key**":	101	111	000	101	111	100	000	101	110	000
"Plaintext":	011	010	100	100	001	010	111	100	000	101
	k	i	l	l	h	i	t	l	e	r

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

One-time Pad

And this is the correct key:

	s	r	l	h	s	s	t	h	s	r
Ciphertext:	110	101	100	001	110	110	111	001	110	101
"Key":	111	101	000	011	101	110	001	011	101	101
"Plaintext":	001	000	100	010	011	000	110	010	011	000
	h	e	l	i	k	e	s	i	k	e

e=000 h=001 i=010 k=011 l=100 r=101 s=110 t=111

Unconditionally secured scheme

For a given ciphertext of same size as the plaintext, there is a equi-probable key that produces it. Thus the scheme is unconditionally secured.

Practical Problems

- Large quantities of random keys are necessary.
- Increases the problem of key distribution.
- 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)

One-time Pad Summary

- Provably secure, when used correctly
  - Cipher-text provides no information about plaintext
  - All plaintexts are equally likely
  - Pad must be random, used only once
  - Pad is known only by sender and receiver
  - Pad is same size as message
  - No assurance of message integrity
- Why not distribute message the same way as the pad?

Entropy Revisited

$P=\{a,b\}$; $p_P(a)=1/4$, $p_P(b)=3/4$
$K=\{K_1,K_2,K_3\}$, $p_K(K_1)=1/2$,
$p_K(K_2)= p_K(K_3)=1/4$

- What is H(P)?
  - $H(P)=(1/4)\log_2(4)+(3/4)\log_2(4/3)\approx0.81$
- $H(K)\approx1.5$
- $H(C)\approx1.85$

Huffman Encoding

- Consider S: a discrete source of symbols
- The messages from S: {s1,s2,…,sk}
- Can we encode these messages such that their average length is as short as possible, and hopefully equal to H(S)?
- Huffman Code provides an optimal solution to this problem.

Informal Description

- The message set X has a probability distribution. Arrange them in ascending order:

  $p(x1) \le p(x2) \le p(x3) \ldots \le p(xj)$
- Initially the codes of each element are empty.
- Choose the two elements with minimum probabilities
- 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.
- When only one element remains, the code of each letter can be constructed by reading the sequence backwards.

Example

- X={a,b,c,d,e}
- p(a)=.05, p(b)=.10, p(c)=.12, p(d)=.13, p(e)=.6

Illustration of the encoding

a	b	c	d	e
.05	**.10**	**.12**	**.13**	**.6**
0	**1**			
.15		**.12**	**.13**	**.6**
		0	**1**	
.15		.25		.6
0		1		
0.4				**1**
0				
1				

x	f(x)
a	000
b	001
c	010
d	011
e	1

Some more results on Entropy

- X and Y are random variables.
  - $H(X,Y) \le H(X)+H(Y)$
- When X and Y are independent:
  - $H(X,Y)=H(X)+H(Y)$
- Conditional Entropy:
  - $H(X|Y)=-\Sigma p(x|y)\log_2 p(x|y)$
- $H(X,Y)=H(Y)+H(X|Y)$
- $H(X|Y) \le H(X)$
  - When X and Y are independent: $H(X|Y)=H(X)$

Theorem

- Let (P,C,K,D,E) be an encryption algorithm. Then
  - $\mathbf{H(K|C)=H(K)+H(P)-H(C)}$
- **Proof:** $H(P,K)=H(C,K)$ [why?]

  or, $H(P)+H(K) = H(K|C)+H(C)$

  or, $\mathbf{H(K|C)=H(K)+H(P)-H(C)}$

Equivocation (ambiguity) of key given the ciphertext

Perfect vs Ideal Ciphers

- **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 as that of the key without the cryptogram.**
- **Such kinds of ciphers are called "ideal ciphers"**
  - **For perfect ciphers, we had H(P)=H(P|C) or, equivalently H(C)=H(C|P)**

Perfect vs Ideal Ciphers

- For perfect ciphers, the key size is infinite if the message size is infinite.
  - however if a shorter key size is used then the cipher can be attacked by someone with infinite computational power.
- Thus, H(K|C) gives us this idea of security (or, insecurity)…

Unicity and Brute Force Attack

- Q: How to protect data against a brute force attacker with infinite computation power?
  - Shannon defined “**unicity distance**” (we shall call it unicity), as the least amount of plaintext which can be deciphered uniquely from the corresponding ciphertext: given unbounded resources by the attacker.
  - Often measured in units of bytes, letters, symbols.

An Important Point

- A common misconception: “any cipher can be attacked by exhaustively trying all possible keys”:
- Thus DES which has a 56 bit key can also be broken by brute force.
  - But if the cipher is used within its unicity then even DES is theoretically secured, like the One Time Pad (OTP).

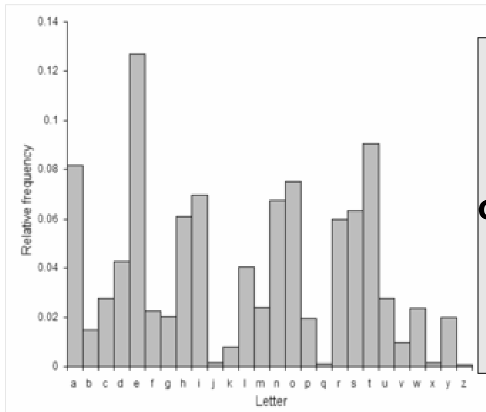
Spurious Keys

- Thus, H(K|C) is the amount of uncertainty that remains of the key after the cipher text is revealed.
  - We know, it is called the key equivocation
- Attacker to guess the key from the ciphertext shall guess the key and decrypt the cipher.
- He checks whether the plaintext obtained is “meaningful” English. If not, he rules out the key.
- But due to the redundancy of language more than one key will pass this test.
- Those keys, apart from the correct key, are called spurious.

Entropy of Plain Text

- H_L: measure of the amount of information per letter of “meaningful” strings of plaintext.
- A random string of plaintext formed using English letter has an entropy of $\log_2|26|\approx 4.76$
- But English letters have a probability distribution.

Frequency of English letters



A first order entropy of the English text is H(P)≈4.76

In general…

- Successive letters have correlation, which reduces the entropy.
- Define P_L to be the random variable that has a probability distribution of n-grams of plaintext
- Define H_L as the **entropy of a natural language** L:

$$H_L = \lim_{n\to\infty} \frac{H(P_n)}{n}$$

Redundancy

Fraction of "excess letters"

Entropy of the language

$$R_L = 1 - \frac{H_L}{\log_2 |P|}$$

Entropy of the random language

For English Language, $1 \le H_L \le 1.5$. Considering $H_L=1.25$, and $|P|=26$, $R_L \approx 0.75$.

English Language is 75% redundant.

A lower Bound of equivocation of key

- P^n: r.v representing n-gram plaintext
- C^n: r.v representing n-gram ciphertext
- $H(K|C^n)=H(K)+H(P^n)-H(C^n)$
  - $H(P^n) \approx nH_L$ (assuming large n)
    $=n(1-R_L)\log_2|P|$
  - $H(C^n) \le n\log_2|C|$
- If $|P|=|C|$,
  - $H(K|C^n) \ge H(K)-nR_L\log_2|P|$

Possible Keys

- Define, K(y)={possible keys given that y is the ciphertext}
  - that is K(y) is the set of those keys for which y is the ciphertext for meaningful plaintexts
- When y is the ciphertext, number of keys is |K(y)|
- Out of them, only one is correct. Rest are spurious.
- So, number of spurious keys=|K(y)|-1

Expected number of spurious keys

- Expected number of spurious keys=average number of spurious keys over all possible ciphertexts is denoted by s_n.

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

Computing the upper bound of equivocation of key

$$
\begin{aligned}
H(K \mid C^n) &= \sum_{y \in C^n} p(y) H(K \mid y) \\
&\le \sum_{y \in C^n} p(y) H(K(y)) \\
&\le \sum_{y \in C^n} p(y) \log_2(|K(y)|) \\
&\le \log_2\Big(\sum_{y \in C^n} p(y)\,|K(y)|\Big) = \log_2(s_n + 1)
\end{aligned}
$$

Lower Bound of spurious keys

- Combining the previous results:

$$
\begin{aligned}
&H(K) - nR_L \log_2 |P| \le \log_2(s_n + 1) \\
&\therefore \log_2(s_n + 1) \ge H(K) - nR_L \log_2 |P|
\end{aligned}
$$

- If the keys are chosen equi-probably: $H(K)=\log_2|K|$. Hence, we have:

$$
\boxed{s_n \ge \frac{|K|}{|P|^{nR_L}} - 1}
$$

Unicity Distance

- Thus increasing n, reduces the number of spurious keys.
- **Unicity Distance** is the number of ciphertexts, n_0 for which the number of spurious keys is reduced to zero.

$$n \geq n_0 = \frac{\log_2 |K|}{R_L \log_2 |P|}$$

This calculation may not be accurate for large values of n

Unicity Distance for Substitution Ciphers

- $|P|=26$
- $|K|=26! \approx 4 \times 10^{26}$, $R_L=0.75$
- $n_0=25$ (approx)
- Given a ciphertext string of length 25, it is possible to predict the correct key uniquely
  - Thus key size alone does not guarantee security, if brute force is possible to an attacker with infinite computational power.

Assignment 1

- Let n be a positive integer. A Latin square of order n is an nxn array L with integers 1,2,…,n such that every integer occurs exactly once in each row and column. An example for n=3 is:

1	2	3
3	1	2
2	3	1

Assignment 1

- Given any Latin square of order n, we can define a related cryptosystem, $e_i(j)=L(i,j)$, where $1 \le i,j \le n$.

  Prove **<u>from the computation of probabilities</u>** that the Latin square cryptosystem achieves perfect secrecy.

Deadline for submission: 20.8.09

Please submit hand written proofs.