Computer Science 418

More on Perfect Secrecy, One-Time Pad, Entropy

Mike Jacobson

Department of Computer Science
University of Calgary

Week 3

Outline

1. Computing $p(C|M)$ and $p(C)$
2. The Vernam One-Time Pad
3. Entropy

Computing $p(C|M)$ and $p(C)$

Recall that perfect secrecy is equivalent to $p(M|C) = p(M)$ for all messages M and all ciphertexts C that occur.

How can we determine $p(C|M)$ and $p(C)$?

For any message $M \in \mathcal{M}$, we have

$$p(C|M) = \sum_{\substack{K \in \mathcal{K} \\ E_K(M)=C}} p(K) \ .$$

That is, $p(C|M)$ is the sum of probabilities $p(K)$ over all those keys $K \in \mathcal{K}$ that encipher M to C.

Number of Keys in the Sum

Usually there is at most one key K with $E_K(M) = C$ for given M and C.

However, some ciphers can transform the same plaintext into the same ciphertext with different keys.

- A monoalphabetic substitution cipher will transform a message into the same ciphertext with different keys if the only differences between the keys occur for characters which do not appear in the message
- Eg. key1 = ECONOMICS, key2 = ECONOMY, and we encrypt a message of at most 6 characters).

Example: Computing $p(C|M)$

$\mathcal{M} = \{a, b\}$, $\mathcal{K} = \{K_1, K_2, K_3\}$, and $\mathcal{C} = \{1, 2, 3, 4\}$. Encryption is given by the following table:

Key	$M = a$	$M = b$
K_1	$C = 1$	$C = 2$
K_2	$C = 2$	$C = 3$
K_3	$C = 3$	$C = 4$

Thus,

$$\begin{array}{ll} p(1|a) = p(K_1)\ , & p(1|b) = 0\ , \\ p(2|a) = p(K_2)\ , & p(2|b) = p(K_1)\ , \\ p(3|a) = p(K_3)\ , & p(3|b) = p(K_2)\ , \\ p(4|a) = 0\ , & p(4|b) = p(K_3)\ . \end{array}$$

Description of E_K

Consider a fixed key K. The mathematical description of the set of all possible encryptions (of any plaintext) under this key K is exactly the image of E_K, *i.e.* the set $E_K(\mathcal{M}) = \{E_K(M) \mid M \in \mathcal{M}\}$.

In the previous example, we have

- $E_{K_1}(\mathcal{M}) = \{1, 2\}$,
- $E_{K_2}(\mathcal{M}) = \{2, 3\}$
- $E_{K_3}(\mathcal{M}) = \{3, 4\}$.

Computation of $p(C)$

For a key K and ciphertext $C \in E_K(\mathcal{M})$, consider the probability $p(D_K(C))$ that the message $M = D_K(C)$ was sent. Then

$$p(C) = \sum_{\substack{K \in \mathcal{K} \\ C \in E_K(\mathcal{M})}} p(K)p(D_K(C))\ .$$

That is, $p(C)$ is the sum of probabilities over all those keys $K \in \mathcal{K}$ under which C has a decryption under key K, each weighted by the probability that that key K was chosen.

Example, cont.

The respective probabilities of the four ciphertexts $1, 2, 3, 4$ are:

$$p(1) = p(K_1)p(a), \quad p(2) = p(K_1)p(b) + p(K_2)p(a)$$
$$p(3) = p(K_2)p(b) + p(K_3)p(a), \quad p(4) = p(K_3)p(b)$$

If we assume that every key and every message is equally probable, *i.e.* $p(K_1) = p(K_2) = p(K_3) = 1/3$ and $p(a) = p(b) = 1/2$, then

$$p(1) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}, \quad p(2) = \frac{1}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{3}$$
$$p(3) = \frac{1}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{3}, \quad p(4) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$$

Note that $p(1|a) = p(K_1) = 1/3 \neq 1/6 = p(1)$, so this system does not provide perfect secrecy.

Necessary Condition for Perfect Secrecy

Theorem 1

If a cryptosystem has perfect secrecy, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Informal argument: suppose $|\mathcal{K}| < |\mathcal{M}|$.

- Then there is some message M such that for a given ciphertext C, no key K encrypts M to C.
- This means that the sum defining $p(C|M)$ is empty, so $p(C|M) = 0$.
- But $p(C) > 0$ for all ciphertexts of interest, so $p(C|M) \neq p(C)$, and hence no perfect security. (The cryptanalyst could eliminate certain possible plaintext messages from consideration after receiving a particular ciphertext.)

Proof of the Theorem

Assume perfect secrecy. Fix a ciphertext C_0 that is the encryption of some message under some key (*i.e.* actually occurs as a ciphertext).

We first claim that for every key K, there is a message M that encrypts to C_0 under key K. Since C_0 occurs as a ciphertext, $C_0 = E_{K_0}(M_0)$ for some key K_0 and message M_0, so by perfect secrecy,

$$\begin{aligned} p(C_0) &= p(C_0|M_0) \\ &= \sum_{\substack{K \in \mathcal{K} \\ E_K(M)=C_0}} p(K) \\ &= p(K_0) + \text{possibly other terms in the sum} \geq p(K_0) > 0\ . \end{aligned}$$

Proof, cont.

Again by perfect secrecy, for every other message $M \in \mathcal{M}$,

$$0 < p(C_0) = p(C_0|M) = \sum_{\substack{K \in \mathcal{K} \\ E_K(M)=C_0}} p(K).$$

In other words, for every M, there is at least one non-zero term in that sum, *i.e.* there exists at least one key K that encrypts M to C_0.

Moreover, different messages that encrypt to C_0 must do so under different keys (as $E_K(M_1) = E_K(M_2)$ implies $M_1 = M_2$). So we have at least as many keys as messages. □

(Formally, consider the set $\mathcal{K}_0 = \{K \in \mathcal{K} \mid C_0 \in E_K(\mathcal{M})\} \subseteq \mathcal{K}$. Then we have shown that the map $\mathcal{K}_0 \to \mathcal{M}$ via $K \mapsto M$ where $E_K(M) = C_0$ is well-defined and surjective. Hence $|\mathcal{K}| \geq |\mathcal{K}_0| \geq |\mathcal{M}|$.)

Shannon's Theorem

Theorem 2 (Shannon's Theorem, 1949/50)

A cryptosystem with $|\mathcal{M}| = |\mathcal{K}| = |\mathcal{C}|$ has perfect secrecy if and only if $p(K) = 1/|\mathcal{K}|$ (i.e. every key is chosen with equal likelihood) and for every $M \in \mathcal{M}$ and every $C \in \mathcal{C}$, there exists a unique key $K \in \mathcal{K}$ such that $E_K(M) = C$.

Proof.

See Theorem 2.8, p. 38, in Katz & Lindell. □

One-Time Pad

Generally attributed to Vernam (1917, WW I) who patented it, but recent research suggests the technique may have been used as early as 1882

- in any case, it was long before Shannon

It is the only substitution cipher that does not fall to statistical analysis.

Bitwise Exclusive-Or

Fix a string length n. Then set $\{0,1\}^n$ is the set of *bit strings* (*i.e.* strings of 0's and 1's) of length$n$.

Definition 1 (bitwise exclusive or, XOR)

For $a, b \in \{0,1\}$, we define

$$a \oplus b = a + b \pmod 2 = \begin{cases} 0 & a = b \, , \\ 1 & a \neq b \, . \end{cases}$$

For $A = (a_1, a_2, \ldots, a_n), B = (b_1, b_2, \ldots, b_n) \in \{0,1\}^n$, we define then

$$A \oplus B = (a_1 \oplus b_1, a_2 \oplus b_2, \ldots, a_n \oplus b_n) \ .$$

(component-wise XOR).

The One-Time Pad

Definition 2 (Vernam one-time pad)

Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$ (bit strings of some fixed length n). Encryption of $M \in \{0,1\}^n$ under key $K \in \{0,1\}^n$ is bitwise XOR, *i.e.*

$$C = M \oplus K \ .$$

Decryption of C under K is done the same way, *i.e.* $M = C \oplus K$, since $K \oplus K = (0, 0, \ldots, 0)$.

Security of the One-Time Pad

Theorem 3

The one-time pad provides perfect secrecy if each key is chosen with equal likelihood. Under this assumption, each ciphertext occurs with equal likelihood (regardless of the probability distribution on the plaintext space).

This means that in the one-time pad, any given ciphertext can be decrypted to *any* plaintext with equal likelihood (defn of perfect secrecy). There is no "meaningful" decryption. So even exhaustive search doesn't help.

Proof of the Theorem

Proof of Theorem 3.

We have $|\mathcal{M}| = |\mathcal{C}| = |\mathcal{K}| = 2^n$, and for every $M, C \in \{0,1\}^n$, there exists a unique key K that encrypts M to C, namely $K = M \oplus C$. By Shannon's Theorem 2, we have prefect secrecy.

Now let $M, C \in \{0,1\}^n$ be arbitrary. Then by perfect secrecy,

$$p(C) = p(C|M) = \sum_{\substack{K \in \{0,1\}^n \\ M \oplus K = C}} p(K)$$

Now $p(K) = 2^{-n}$ for all keys K, and the sum only has one term (corresponding to the unique key $K = M \oplus C$). Hence $p(C) = 2^{-n}$ for every $C \in \{0,1\}^n$. □

Cryptanalysis of the One-Time Pad

It is imperative that each key is only used once:

- Immediately falls to a KPA: if a plaintext/ciphertext pair (M, C) is known, then the key is $K = M \oplus C$.
- Suppose K were used twice:

  $$C_1 = M_1 \oplus K \ , C_2 = M_2 \oplus K \implies C_1 \oplus C_2 = M_1 \oplus M_2 \ .$$

  Note that $C_1 \oplus C_2 = M_1 \oplus M_2$ is just a coherent running key cipher (adding two coherent texts, M_1 and M_2), which as we have seen is insecure.

For the same reason, we can't use shorter keys and "re-use" portions of them. Keys must be randomly chosen and at least as long as messages. This makes the one-time pad impractical.

Practical Issues

Main disadvantages of one-time pad:

- requires a random key which is as long as the message
- each key can be used only once.

One-time schemes are used when perfect secrecy is crucial and practicality is less of a concern, for example, Moscow-Washington hotline.

One-Time Pad: Conclusion

The major problem with the one-time pad is the cost. As a result, we generally rely on *computationally secure* ciphers.

- These ciphers would succumb to exhaustive search, because there is a unique "meaningful" decipherment.
- The computational difficulty of finding this solution foils the cryptanalyst.
- A *proof* of security does not exist for any proposed computationally secure system.

Measuring Information

Recall that information theory captures the amount of information in a piece of text.

Measured by the average number of bits needed to encode all possible messages in an *optimal prefix-free* encoding.

- optimal – the average number of bits is as small as possible
- prefix-free – no code word is the beginning of another code word (*e.g.* can't have code words 01 and 011 for example)

Formally, the amount of information in an outcome is measured by the *entropy* of the outcome (function of the probability distribution over the set of possible outcomes).

Example

The four messages

UP, DOWN, LEFT, RIGHT

could be encoded in the following ways:

String	Character	Numeric	Binary
"UP"	"U"	1	00
"DOWN"	"D"	2	01
"LEFT"	"L"	3	10
"RIGHT"	"R"	4	11
(40 bits)	(8 bits)	(16 bits)	(2 bits)
(5 char string)	8-bit ASCII	(2 byte integer)	2 bits

Coding Theory

In the example, all encodings carry the same information (which we will be able to measure), but some are more efficient (in terms of the number of bits required) than others.

Note: *Huffmann encoding* can be used to improve on the above example if the directions occur with different probabilities.

This branch of mathematics is called *coding theory* (and has nothing to do with the term "code" defined previously).

Entropy

Definition 3

Let X be a random variable taking on the values $X_1, X_2, \ldots, X_n$ with a probability distribution

$$p(X_1), p(X_2), \ldots, p(X_n) \quad \text{where} \quad \sum_{i=1}^{n} p(X_i) = 1$$

The *entropy* of X is defined by the weighted average

$$H(X) = \sum_{\substack{i=1 \\ p(X_i)\neq 0}}^{n} p(X_i) \log_2 \frac{1}{p(X_i)} = - \sum_{\substack{i=1 \\ p(X_i)\neq 0}}^{n} p(X_i) \log_2 p(X_i) \ .$$

Intuition

- An event occurring with prob. 2^{-n} can be optimally encoded with n bits.
- An event occurring with probability p can be optimally encoded with $\log_2(1/p) = -\log_2(p)$ bits.
- The weighted sum $H(X)$ is the expected number of bits (*i.e.* the amount of information) in an optimal encoding of X (*i.e.* one that minimizes the number of bits required).
- If $X_1, X_2, \ldots, X_n$ are outcomes (e.g. plaintexts, ciphertexts, keys) occurring with respective probabilities $p(X_1), p(X_2), \ldots, p(X_n)$, then $H(X)$ is the amount of information conveyed about these outcomes.

Example 1

Suppose $n > 1$ and $p(X_i) > 0$ for all i. Then

$$\begin{aligned} 0 < p(X_i) < 1 \quad (i = 1, 2, \ldots, n) \\ \frac{1}{p(X_i)} > 1 \\ \log_2 \frac{1}{p(X_i)} > 0, \end{aligned}$$

hence $H(X) > 0$ if $n > 1$.

If there are at least 2 outcomes, both occurring with nonzero probability, then either one of them conveys information.

Example 2

Suppose $n = 1$. Then

$$p(X_1) = 1, \quad \frac{1}{p(X_1)} = 1, \quad \log_2 \frac{1}{p(X_1)} = 0 \implies H(X) = 0 \ .$$

One single possible outcome conveys no new information (you already know what it's going to be).

In fact, for arbitrary n, $H(X) = 1$ if and only of $p_i = 1$ for exactly one i and $p_j = 0$ for all $j \neq i$.

Example 3

Suppose there are two possible outcomes which are equally likely:

$$\begin{aligned} p(\text{heads}) = p(\text{tails}) = \frac{1}{2}, \\ H(X) = \frac{1}{2}\log_2 2 + \frac{1}{2}\log_2 2 = 1 \ . \end{aligned}$$

Seeing either outcome conveys exactly 1 bit of information (heads or tails).

Example 4

Suppose we have

$$p(UP) = \frac{1}{2}, \quad p(DOWN) = \frac{1}{4}, \quad p(LEFT) = \frac{1}{8}, \quad p(RIGHT) = \frac{1}{8} \ .$$

Then

$$\begin{aligned} H(X) &= \frac{1}{2}\log_2 2 + \frac{1}{4}\log_2 4 + \frac{1}{8}\log_2 8 + \frac{1}{8}\log_2 8 \\ &= \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{3}{8} = \frac{14}{8} = \frac{7}{4} = 1.75 \ . \end{aligned}$$

An optimal prefix-free (Huffman) encoding is

$$UP = 0, \quad DOWN = 10, \quad LEFT = 110, \quad RIGHT = 111 \ .$$

Because UP is more probable than the other messages, receiving UP conveys less information than receiving one of the other messages. The *average* amount of information received is 1.75 bits.

Example 5

Suppose we have n outcomes which are equally likely: $p(X_i) = 1/n$.

$$H(X) = \sum_{i=1}^{n} \frac{1}{n}\log_2 n = \log_2 n \ .$$

So if all outcomes are equally likely, then $H(X) = \log_2 n$.

If $n = 2^k$ (*e.g.* each outcome is encoded with k bits), then $H(X) = k$.

Application to Cryptography

For plaintext space $\mathcal{M}$, $H(\mathcal{M})$ measures the uncertainty of plaintexts.

Gives the amount of partial information that must be learned about a message in order to know its whole content when it has been

- distorted by a noisy channel (coding theory) or
- hidden in a ciphertext (cryptography)

For example, consider a ciphertext C = X$7PK that is known to correspond to a plaintext $M \in \mathcal{M} = \{$ "heads", "tails" $\}$.

- $H(\mathcal{M}) = 1$, so the cryptanalyst only needs to find the distinguishing bit in the first character of M, not all of M.

Maximal Entropy

Recall that the entropy of n equally likely outcomes (*i.e.* each occurring with probability $1/n$) is $\log_2(n)$. This is indeed the maximum:

Theorem 4

$H(X)$ is maximized if and only if all outcomes are equally likely. That is, for any n, $H(X) = \log_2(n)$ is maximal if and only if $p(X_i) = 1/n$ for $1 \le i \le n$. $H(X) = 0$ is minimized if and only if $p(X_i) = 1$ for or exactly one i and $p(X_j) = 0$ for all $j \ne i$.

Intuitively, $H(X)$ decreases as the distribution of messages becomes increasingly skewed.

Idea of Proof

(This is *not* a complete proof! The full proof uses induction on n or Jensen's inequality.)

Idea.

Suppose $p(X_1) > 1/n$, $p(X_2) < 1/n$, and $p(X_i)$ is fixed for $i > 2$. Set $p = p(X_1)$, then $p(X_2) = 1 - p - \epsilon$ with $\epsilon = p(X_3) + \cdots + p(X_n)$, and

$$\begin{aligned} H &= -p\log(p) - (1 - p - \epsilon)\log(1 - p - \epsilon) - \epsilon)\log(\epsilon) \\ \frac{dH}{dp} &= -\log p - 1 + \log(1 - p - \epsilon) + 1 \\ &= \log\frac{1 - p - \epsilon}{p} < 0 \quad \text{since } 0 < p < 1 \end{aligned}$$

Thus, H decreases as a function of p as p increases. Now prove that H is maximized when $p = 1/n$ by induction on n. □

Notes

For a key space $\mathcal{K}$, $H(\mathcal{K})$ measures the amount of partial information that must be learned about a key to actually uncover it (*e.g.* the number of bits that must be guessed correctly to recover the whole key).

For a k bit key, the best scenario is that all k bits must be guessed correctly to know the whole key (*i.e.* no amount of partial information reveals the key, only full information does).

- Entropy of the random variable on the key space shoul be maximal.
- Previous theorem: happens exactly when each key is equally likely.
- Best strategy to select keys in order to give away as little as possible is to choose them with equal likelihood (*uniformly at random*).

Cryptosystems are assessed by their key entropy, which ideally should just be the key length in bits (*i.e.* maximal).