CS3381-Cryptography

Lecture 3: One-time Pad and Perfect Secrecy

January 24, 2017

By tweaking the Vigenère cipher in a simple way, we obtain an encryption method called the *one-time pad*, or *Vernam cipher.* We show the following properties of the one-time pad.

- The one-time pad possesses a special property called *perfect secrecy*. Essentially this means that an attacker obtains *no* information about the plaintext from the ciphertext.
- Perfect secrecy notwithstanding, the one-time pad has several serious defects: a key can never be reused. Furthermore, messages encrypted with a one-time pad are *malleable*: an attacker can meaningfully alter a message even if they are unable to decrypt the entire message.
- Perfect secrecy comes at a price: We prove that perfect secrecy requires the key to be as long as the plaintext.

1 The one-time pad.

This is essentially the Vigenère cipher, or rather the binary version of the Vigenére cipher, in which the key block does not repeat: thus the key has the same length as the message. Formally,

$$\mathcal{K} = \mathcal{M} = \{0,1\}^n$$

for some $n > 0$. That is, both the key and the message are bit strings of some fixed length n. We set for any $k, m \in \{0,1\}^n$,

$$E(k,m) = k \oplus m.$$

As with the Vigenère cipher, the decryption function is identical to the encryption function.

Example. Suppose the plaintext message p is

`Attack!`

and the key k is

`Win@War`

Note that both strings are 7 characters long, so in this case, $n = 7 \times 8 = 56$. These ASCII strings are encoded as sequences of bytes, and the ciphertext is

$$c = E(k, p) = p \oplus k.$$

The resulting sequence of bytes is not printable when represented as an ASCII string, but we can represent it in hex as

`16 1d 1a 21 34 0a 53`

Now suppose we have a second plaintext message p_2:

`Retreat`

Set

$$k' = c \oplus p_2 = k \oplus p_1 \oplus p_2.$$

(Incidentally, in this example k' is `DxnSQk'`.) Then

$$E(k', p_2) = k' \oplus p_2 = c \oplus p_2 \oplus p_2 = c.$$

In other words, on intercepting the ciphertext c, the attacker cannot tell whether the plaintext is `Attack!`, or `Retreat` or, for that matter, any other 7-byte string. This is what we mean, roughly, when we say that the attacker obtains no information about the plaintext from the ciphertext.

Observe that while the original key k is printable and in more or less normal English, the new key is likely to be neither. If the attacker can expect that intelligible keys will be used, then the system is no longer perfectly secret, as certain plaintexts can then be ruled out. It is important that the key be chosen uniformly at random from the set of all possible keys.

2 Perfect secrecy formally defined

It's not really true that the one-time pad reveals *no* information about plaintexts from ciphertexts alone: If we use this scheme to encrypt messages of various lengths, the length of the ciphertext will always tell us the length of the plaintext.

If you think about it, this is true of any cryptographic system that allows encryption of messages of arbitrary length: Long messages cannot have short encryptions. (See the homework exercises for more precision on this point.) Thus when we say that ciphertext reveals no information about plaintext, we really mean to restrict to systems where all the messages have the same length.

So consider a cryptographic system where $\mathcal{M} = \{0,1\}^r$. Fix a pair of elements $m, c \in \mathcal{M}$. How likely is it that c is the encryption of m? We denote this probability

$$\Pr_{k \in \mathcal{K}}(c = E_k(m)).$$

The underlying probability space is the set of all keys in $\mathcal{K}$ with the uniform distribution. The probability is accordingly

$$\frac{|\{k \in \mathcal{K} : c = E_k(m)\}|}{|\mathcal{K}|}.$$

We say the system has *perfect secrecy* if for all $m_1, m_2, c \in \mathcal{M}$,

$$\Pr_{k \in \mathcal{K}}(c = E_k(m_1)) = \Pr_{k \in \mathcal{K}}(c = E_k(m_2)).$$

Equivalently,

$$|\{k \in \mathcal{K} : c = E_k(m_1)\}| = |\{k \in \mathcal{K} : c = E_k(m_2)\}|.$$

(Again, remember that we are assuming all messages have the same length.)

3 The one-time pad has perfect secrecy

The proof that the one-time pad has perfect secrecy is trivial: For any $m, c \in \{0,1\}^n$, there is exactly one k such that $E_k(m) = c$, namely

$$k = c \oplus m.$$

Thus for any $m_1, m_2, c \in \mathcal{M}$,

$$|\{k \in \mathcal{K} : c = E_k(m_1)\}| = 1 = |\{k \in \mathcal{K} : c = E_k(m_2)\}|.$$

4 What is 'one-time' about the one-time pad?

If we use the one-time pad to encrypt several different messages with the same key, then a great deal of information is leaked. For example, if $m_1, \ldots, m_q$ are ordinary English text in ASCII, all of the same length, and

$$c_i = k \oplus m_i$$

for all i, then we can carry out the attack of Assignment 1: The set of first bytes of all q of the c_i has an English-like distribution, and by trying out all 256 possibilities for the first byte of k, we recover the first byte of all of the m_i. We recover all the other plaintext bytes in the same manner. In fact, using the one-time pad repeatedly with the same key k is effectively the Vigenère cipher with key k, where the plaintext is the concatenation of all the plaintexts m_i. This will work as long as q is not too short, $q = 20$ or 30 is probably sufficient.

Even if we only use the one-time pad *twice,* that is, if we take $q = 2$, some information is still leaked. We have $c_1 \oplus c_2 = m_1 \oplus m_2$, so at the very least, we can determine all the bytes where the plaintexts m_1, m_2 agree and where they disagree. In fact, if m_1, m_2 are long enough messages in English, then it is possible to recover the complete plaintext.

5 Malleability

The attacker Eve can meaningfully alter an intercepted ciphetext, even if she cannot decrypt it. Let us suppose, for example, that we know, or strongly suspect, that plaintext of the intercepted message begins with a standard header:

```
'FROM: Alice \n'
```

There are thirteen characters, including spaces and the newline character, in the message above. We can use this together with the ciphertext to recover the first eleven bytes k' of the key k. This gives us absolutely no help in deciphering what the message from Alice actually is. However, we now have enough information to modify the first few bytes of the ciphertext, so that when the altered message is decrypted with k, it will now begin:

```
'FROM: Joe   \n'
```

(Two additional spaces have been inserted.) Eve can forward the modified ciphertext to the intended recipient, who will now believe that it was sent from Joe.

In general, a secure system for encryption and decryption does nothing to guarantee authenticity of the received message. We will study the problem of message authentication later.

6 Perfect secrecy requires keys as long as the messages

The inconvenience of the one-time pad comes at a price. We have to have as many bytes of key as bytes we are encrypting, so both sender and recipient have to be equipped with an essentially unlimited supply of randomly-generated bytes in order to encrypt and decrypt a large volume of traffic. We prove here that this is unavoidable:

Theorem 1 *In a perfectly secret cryptosystem with keyspace $\mathcal{K}$ and message space $\mathcal{M}$,*

$$|\mathcal{K}| \geq |\mathcal{M}|.$$

For example, if we are encrypting n-bit messages, then $|\mathcal{M}| = 2^n$. The theorem tells us that there must be *at least* 2^n keys, and thus if we use fixed-length bit strings as keys, keys must be at least n bits long, which is exactly what we see in the one-time pad.

Proof. Suppose $|\mathcal{K}| < |\mathcal{M}|$. We will show that the system is not perfectly secret. Pick $c \in \mathcal{M}$, and let

$$\mathcal{U} = \{D_k(c) : k \in \mathcal{K}\}.$$

Then $|\mathcal{U}| \leq |\mathcal{K}| < |\mathcal{M}|$, so $\mathcal{U} \subsetneq \mathcal{M}$. Thus we can pick $m_1 \in \mathcal{U}$ and $m_2 \in \mathcal{M} \backslash \mathcal{U}$. We then have

$$|\{k \in \mathcal{K} : c = E_k(m_1)\}| \geq 1,$$

and

$$|\{k \in \mathcal{K} : c = E_k(m_2)\}| = 0,$$

so we cannot have perfect secrecy.

7 An equivalent definition of perfect secrecy: Indistinguishability experiment.

We give an equivalent definition of perfect secrecy. This is more complicated than our original definition, but we will later see that by tweaking this definition, we will arrive at a definition of security for systems that do not possess perfect secrecy.

We describe a probabilistic experiment involving two parties, the *adversary* and the *challenger*: The adversary claims that the system is insecure, and the challenger challenges the adversary to demonstrate this. The experiment proceeds as follows:

- The challenger chooses a key $k \in \mathcal{K}$ uniformly at random, and, independently, a bit $b \in \{0,1\}$ uniformly. (These are *not* communicated to the challenger.)
- The adversary chooses two messages $m_0, m_1 \in \mathcal{M}$, with $|m_0| = |m_1|$ (equal length), and sends these to the challenger.
- The challenger computes $c = E_k(m_b)$, and sends it to the adversary.
- The adversary computes a bit $b' \in \{0,1\}$ and claims that c is the encryption of $m_{b'}$.

The adversary *succeeds* if $b = b'$. What is the probability of success?

Theorem 2 *The cryptographic system is perfectly secret if and only if for every $m_0, m_1 \in \mathcal{M}$ with $|m_0| = |m_1|$, and every method of computing b', the adversary's probability of success is exactly $\frac{1}{2}$.*

That is, the adversary can do no better than flipping a coin, even if he gets to choose the plaintexts m_0, m_1.

Proof. First, suppose that the system is perfectly secret. The adversary picks m_0, m_1. Let us suppose that his method is to answer $b' = 0$ if the ciphertext c sent by the challenger satisfies

$$c \in \{c_1, \ldots, c_r\},$$

and $b' = 1$ if

$$c \in \{c'_1, \ldots, c'_s\},$$

where

$$\mathcal{M} = \{c_1, \ldots, c_r\} \cup \{c'_1, \ldots, c'_s\}.$$

The adversary succeeds if $b = 0$ and $E_k(m_0) \in \{c_1, \ldots, c_r\}$, or if $b = 1$ and $E_k(m_1) \in \{c'_1, \ldots, c'_s\}$. The resulting probability is

$$\frac{1}{2} \cdot \sum_{i=1}^{r} \Pr_{k \in \mathcal{K}}(E_k(m_0) = c_i) + \frac{1}{2} \cdot \sum_{j=1}^{s} \Pr_{k \in \mathcal{K}}(E_k(m_1) = c'_j).$$

Since the system is perfectly secret, we can replace m_0 in the first summand above by m_1, which makes the success probability

$$\begin{aligned}
\frac{1}{2} \cdot \sum_{i=1}^{r} \Pr_{k \in \mathcal{K}}(E_k(m_1) = c_i) + \frac{1}{2} \cdot \sum_{j=1}^{s} \Pr_{k \in \mathcal{K}}(E_k(m_1) = c'_j) &= \\
\frac{1}{2} \cdot \Pr_{k \in \mathcal{K}}(E_k(m_1) \in \{c_1, \ldots, c_r\} \cup \{c'_1, \ldots, c'_s\}) &= \\
\frac{1}{2} \cdot \Pr_{k \in \mathcal{K}}(E_k(m_1) \in \mathcal{M}) &= \\
\frac{1}{2} \cdot 1 &= \\
\frac{1}{2}.
\end{aligned}$$

For the converse direction, assume that the system is *not* perfectly secret. Then there is some pair m_0, m_1 of plaintexts and some ciphertext c' such that

$$\Pr_{k \in \mathcal{K}}(E_k(m_0) = c') \neq \Pr_{k \in \mathcal{K}}(E_k(m_1) = c').$$

Let's assume without loss of generality that the left-hand side is strictly larger than the right-hand side. We will show that the adversary has a guessing strategy that succeeds with probability strictly greater than one-half. The adversary provides m_0, m_1 to the challenger, and uses the following guessing strategy: If c is a ciphertext such that

$$\Pr_{k \in \mathcal{K}}(E_k(m_0) = c) > \Pr_{k \in \mathcal{K}}(E_k(m_1) = c),$$

then the adversary guesses that the plaintext is m_0, otherwise the adversary guesses m_1. As above, the success probability is

$$\frac{1}{2} \cdot \sum_{i=1}^{r} \Pr_{k \in \mathcal{K}}(E_k(m_0) = c_i) + \frac{1}{2} \cdot \sum_{j=1}^{s} \Pr_{k \in \mathcal{K}}(E_k(m_1) = c'_j),$$

where $\{c_1, \ldots, c_r\}$ is the set of ciphertexts for which the adversary guesses m_0, and $\{c'_1, \ldots, c'_s\}$ those for which the adversary guesses m_1. But our assumption implies that this is strictly greater than

$$\frac{1}{2} \cdot \sum_{i=1}^{r} \Pr_{k \in \mathcal{K}}(E_k(m_1) = c_i) + \frac{1}{2} \cdot \sum_{j=1}^{s} \Pr_{k \in \mathcal{K}}(E_k(m_1) = c'_j) = \frac{1}{2},$$

thus the system does not satisfy our second alternative definition. This completes the proof.

(In the proof we assumed that the adversary's method was *deterministic*: that is, for every possible ciphertext $c \in \mathcal{M}$, the adversary computes a fixed answer that depends only on c, m_0, m_1. But we can also allow the adversary to flip some coins during the computation, and thus use a *probabilistic* method. In this case, the probability in the altered definition is over all possible keys, the challenger's coin flip, and the adversary's coin flip. This makes the proof a little more complicated, but the result is the same.)

8 If you can't have perfect secrecy, what can you have?

The problem with using perfect secrecy as a definition of security is that it is much too stringent: it requires our system to be secure against an infinitely powerful adversary: We place no limits on the amount of work that the adversary has to do to find messages m_0 and m_1 that break the system, nor on the work required to compute the guess b' from the challenge ciphertext c. We also require the adversary to gain absolutely no advantage over just random guessing.

If the adversary has to perform 2^{100} operations to find m_0 and m_1 or to compute b', or if the probability of success is only $\frac{1}{2} + 2^{-100}$, then in practical terms it is still impossible for the adversary to get any useful information from the ciphertext. Thus our definition of security will restrict the adversary to performing only *feasible* computations, and require that the advantage over a success probability of $\frac{1}{2}$ is more than *negligible.* We will flesh this out, and provide exact definitions of 'feasible' and 'negligible' a bit later.