Introduction to Cryptography January 18, 2018

Lecture 2: Classical Ciphers and Perfect Secrecy

Instructor: Vipul Goyal *Scribe: David Edelstein*

1 Symmetric Key Cryptography

$G(\cdot) \rightarrow k$
$Enc(m, k) \rightarrow CT$
$Dec(CT, k) \rightarrow m$
In other words, the key the generator gives you is used for both encrypting and decrypting messages

Definition 1 *A symmetric key cryptographic scheme is said to have the property of correctness if it is guaranteed that encrypting a message with a key and then decrypting using that same key will produce the original message. Formally, for key k and message m, that: $m = Dec(Enc(m, k), k$*

2 Caesar Cipher

The Caesar cipher is **one of the most ancient ciphers**, named after Julius Caesar, who more than 2000 years ago is said to have used it to convey secret messages, and it likely even predates him.

In it, **letters of the alphabet are rotated some number**. For instance, in a Caesar cipher with shift of 1, the message ATTACK would become BUUBDL, since B is the letter after A, U is the letter after T, and so on. The alphabet is considered to loop around, so Z would become A with a shift of one. More formally, the generator $G(\cdot)$ produces a single letter of the alphabet which A will map to, which defines the shift.

The **Caesar cipher is decoded by reversing the shift**. For instance, given the cipher text IJKJSI with shift 5, I may decode the message to DEFEND, because D is five letters before I, E is five letters before J, and so on.

Try it out! Encode JULIUS with a shift of 9, and decode EYKYWXYW from its shift of four. As another bit of practice, try this puzzle I wrote that uses Caesar shifts. The answer is ultimately one word — can you figure it out?

The Caesar cipher is **easily broken simply by trying all keys** — in English, there's only 26. It's not very secure, though that hasn't stopped it from being used by individuals up to today. One popular version is ROT13, in which letters are rotated by 13, with A becoming N and so on.

3 Substitution Cipher

In a substitution cipher, each letter is mapped to another, but unlike in the Caesar cipher, the transformation is not uniform. Instead of producing a single letter giving a shift, **the generator function in a substitution cipher yields a permutation of the alphabet**. This serves as

a lookup table. **To encode a message, map each character of the message to its corresponding letter in the lookup table. To decode, simply apply the lookup table in reverse.**

Substitution ciphers are much harder to crack than Caesar ciphers, because there's too many possible keys ($26! \approx 4 * 10^{23}$) to try them all. But with a large enough corpus and a bit of guesswork, they can be cracked using frequency analysis attacks. In English, for instance, here are the three most common letters and their frequency:

E — 12.5%
T — 9.28%
A — 8.09%

So with a large text encoded using a substitution cipher, it's reasonable to guess that the most-used letter is actually E, and that the next two are probably T and A.

Bigrams (sequences of two characters) also have known frequencies. Double letters are a special case of bigrams that can be particularly useful. Here are some sample bigram frequencies:

TH — 3.56%
IN — 2.43%
EE — 0.38%
OO — 0.21%

Using these, a substitution cipher may be gradually unravelled. For more information about frequency attacks, read this post.

Remark 1 *Substitution ciphers may also map to a symbol set other than the alphabet, such as* $A \rightarrow ☹$. *Caesar ciphers are a special case of substitution ciphers. There are many variations of substitution ciphers, but all are fundamentally vulnerable to frequency attacks.*

4 Vigenère Cipher

A Vigenère cipher involves different Caesar shifts being applied to subsets of the letters of a message. It was first described in the mid 1500s by Giovan Battista Bellaso, but credit went to Blaise de Vigenère anyhow. It was long considered the gold standard of ciphers.

Key generation produces a short random string. To encode a message, **replicate the key out to the length of the message, then shift each letter an amount corresponding to the letter of the key**. To decode it, again **replicate the key and apply the shifts in reverse**.

For example, to encrypt CRYPTOGRAPHIC with the key BEZ, replicate the key as follows:

```
CRY PTO GRA PHI C
BEZ BEZ BEZ BEZ B
```

Then apply the shifts, producing:

```
DVX QXN HVZ QLH D
```

Which curiously avoids having any vowels.

Cracking a Vigenère cipher **starts with figuring out the length of the key**. There are several

heuristic methods for this, such as checking factors of the distance between chunks of repeated cipher text. Once the key length is found, **the differently-shifted subsets are susceptible to frequency analysis**, which is even more effective than against substitution ciphers.

If the key length reaches that of the message, the Vigenère cipher reduces to a one-time pad. But before we can discuss that, let's define a new concept...

5 Perfect Security

What would it mean for a cryptographic scheme to be secure? Consider an adversary $\mathcal{A}$ who has cipher text c (from the space of cipher texts $\mathcal{C}$) and wishes to find the original message m (from the space of messages $\mathcal{M}$) but does not have the key k (from the space of keys $\mathcal{K}$).

Would the following be reasonable conditions for security?

$\mathbb{P}[\mathcal{A}(c) \to m] = 0$

No. This is impossible; the adversary can always guess the message.

$\mathbb{P}[\mathcal{A}(c) \to m]$ is very, very small

No. Assurances that they won't guess the whole message are insufficient when it's possible for them to still decrypt vital portions of it.

$\mathbb{P}[\mathcal{A}(c)$ guesses any given character$] = 1/26$

No. This is too optimistic. Frequency analysis can let the adversary get individual characters with higher confidence than chance. Conversely, this doesn't rule out the adversary still being able to learn things about the message as a whole that don't have to do with individual characters.

A better definition is that **having c doesn't give the adversary any information about** m. c is independent from m. Put formally:

Definition 2 *If a cryptographic scheme meets the following condition:*

$$\forall (m_1, m_2) \in \mathcal{M}, \forall c \in \mathcal{C} : \mathbb{P}[k \leftarrow G(\cdot) : Enc(m_1, k) = c] = \mathbb{P}[k \leftarrow G(\cdot) : Enc(m_2, k) = c]$$

Then it is perfectly secure.

Note that perfect security is only over a domain of messages. **The adversary may still know certain things about the encrypted message from their knowledge of** $\mathcal{M}$, and indeed such information is impossible to conceal. For instance, if $\mathcal{M}$ only contains messages of up to 100 characters, then the adversary will still know of a perfectly secure message that its length is no more than 100 characters.

6 One-Time Pad

This is not merely a theoretical possibility. There exists an encryption scheme, discovered more than a hundred years ago, called the one-time pad, **which is perfectly correct and perfectly secure**.

Definition 3 *A cryptographic scheme that is both correct and perfectly secure is called perfectly correct.*

To use the one-time pad, **the key generator produces a random string of length at least that of the message**. To encrypt, **the key is xored with the message**. To decrypt, **the key is just xored with the cipher text**. Formally:

$G(\cdot) \to k$, where k is a random string at least as long as any message to be encrypted
$Enc(m, k) \to m \oplus k = c$
$Dec(c, k) \to c \oplus k = m$

A one-time pad is perfectly correct.

Proof. Because $((x \oplus y) \oplus y) = x)$ for any x and y, **a one-time pad is always correct**.
Given a cipher $c \in \mathcal{C}$ and a message length n, $\forall m \in \mathcal{M}$

$\mathbb{P}[k \leftarrow G(\cdot) : Enc(m, k) = c]$
$= \mathbb{P}[k \leftarrow G(\cdot) : m \oplus k = c]$
$= \mathbb{P}[k \leftarrow G(\cdot) : c \oplus m = k]$
$= 2^{-n}$ because the key bits are random and so each of the n bits has a $1/2$ chance of matching the corresponding bit of $c \oplus m$.

All messages then have the have the same likelihood given a certain cipher text, so the adversary gains no information about the message from seeing the cipher text and **the condition for perfect security is met**. Since the scheme is always correct and is perfectly secure, **one-time pads are perfectly correct**. ■

7 Shannon's Theorem

Shannon's theorem tells us that there aren't really any other perfectly correct cryptographic schemes.

Theorem 1 (Shannon's theorem) *It is impossible to have a perfectly correct symmetric key scheme if $|\mathcal{K}|$ (the number of possible keys) is less than $|\mathcal{M}|$.*

Proof. Consider a scheme for which $|\mathcal{K}| < |\mathcal{M}|$. We will show that $\exists c, m_1, m_2$ such that the perfect security condition is violated.
Generate all $|\mathcal{K}|$ possible keys $\{k_1, k_2, ..., k_{|\mathcal{K}|}\}$
For a cipher text $c = Enc(m_1, k_1)$, attempt to decrypt it with each of those keys:

$s_1 = Dec(c, k_1)$
$\vdots$
$s_{|\mathcal{K}|} = Dec(c, k_{|\mathcal{K}|})$

And gather these attempts together into a set S.
$|S| \leq |\mathcal{K}| < |\mathcal{M}|$ because each key produces one no more than one unique decryption attempt.
Because $|S| < |\mathcal{M}|$, $\exists m_2 \in \mathcal{M}$ s.t. $m_2 \notin S$ and for this m_2

$\mathbb{P}[k \leftarrow G(\cdot) : Enc(m_2, k) = c] = 0 \neq \mathbb{P}[k \leftarrow G(\cdot) : Enc(m_1, k) = c]$

Therefore the perfect security condition is violated and this scheme is not perfectly correct. ■

8 Limits of the One-Time Pad

If the one-time pad is really always correct and perfectly secure, then what are we doing with the next twelve weeks of this class? The problem is that while a one-time pad is perfect and unlike most mathematically perfect things, it's actually possible in the real world, it's still usually not very practical.

The key must be as long as the message, or else we're back to the eminently crackable Vigenère cipher. For many things we'd like to send, the message may be very large, and so the key is much harder to surreptitiously hand over, store, and eventually destroy than a memorizable Vigenère key or the alphabetic permutation of a substitution cipher.

Generating a truly random key is also difficult and expensive, and historic one-time pads have fallen to careful cryptanalysis of the pseudorandom number generators behind the keys.

For most applications, then, the one-time pad is impractical and people turn to cryptographic schemes which aren't perfectly secure, but which are provably secure. It's those we'll be learning about for the rest of the semester.