CPSC 467b: Cryptography and Computer Security

Michael J. Fischer

Lecture 3
January 13, 2012

Perfect secrecy

Caesar cipher

Recall: Base Caesar cipher:

$$E_k(m) = (m + k) \bmod 26$$

$$D_k(c) = (c - k) \bmod 26.$$

Full Caesar cipher:

$$E_k^r(m_1 \dots m_r) = E_k(m_1) \dots E_k(m_r)$$

$$D_k^r(c_1 \dots c_r) = D_k(c_1) \dots D_k(c_r).$$

Simplified Caesar cipher

A probabilistic analysis of the Caesar cipher.

Simplify by restricting to a 3-letter alphabet.

$\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0, 1, 2\}$
$E_k(m) = (m + k) \bmod 3$
$D_k(m) = (m - k) \bmod 3.$

Theorem

The simplified Caesar cipher achieves perfect secrecy.

Joint message-key distribution

A priori message probabilities:

m	p_m
0	1/2
1	1/3
2	1/6

Each key has probability 1/3.

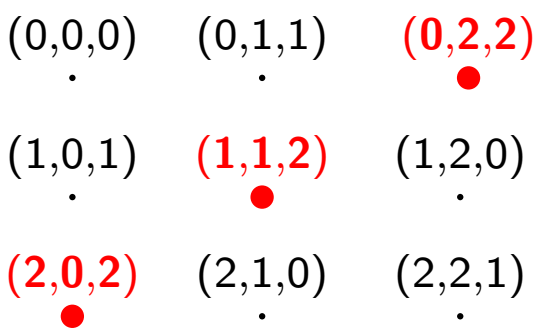
Joint probability distribution:

m \ k	0	1	2
0	1/6	1/6	1/6
1	1/9	1/9	1/9
2	1/18	1/18	1/18

Conditional probability distribution

$\Pr[\mathrm{m} = 1] = 1/3$.
Eve sees $c = 2$.
She wishes to compute $\Pr[\mathrm{m} = 1 \mid \mathrm{c} = 2]$.

First, find the sample space Ω.
Points in Ω are triples (m, k, c), where $c = E_k(m)$.:



Points for which $c = 2$ are shown in bold red.

Proof of perfect secrecy

$\Pr[c = 2]$ is the sum of the probabilities of the bold face points, i.e., $1/6 + 1/9 + 1/18 = 6/18 = 1/3$.

		k = 0	1	2
m	0	1/6	1/6	**1/6**
	1	1/9	**1/9**	1/9
	2	**1/18**	1/18	1/18

The only point for which $m = 1$ is $(1, 1, 2)$ (the center point).
It's probability is $1/9$, so $\Pr[m = 1 \wedge c = 2] = 1/9$.
By definition of conditional probability,

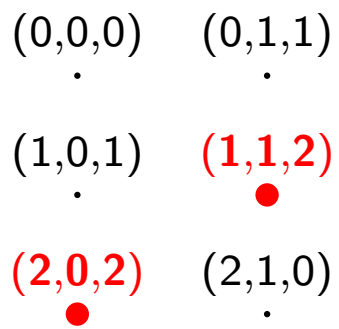
$$\Pr[m = 1 \mid c = 2] = \frac{\Pr[m = 1 \wedge c = 2]}{\Pr[c = 2]} = \frac{1/9}{1/3} = \frac{1}{3} = \Pr[m = 1].$$

Similarly, $\Pr[m = m_0 \mid c = c_0] = \Pr[m = m_0]$ for all m_0 and c_0.
Hence, simplified Caesar cipher is information-theoretically secure.

A minor change

Suppose we reduce the key space to $\mathcal{K} = \{0, 1\}$.
The a priori message distribution stays the same, but the joint probability distribution changes as does the sample space.

m \ k	0	1
0	1/4	1/4
1	1/6	1/6
2	1/12	1/12



Now, $\Pr[\mathrm{c} = 2] = 1/6 + 1/12 = 3/12 = 1/4$, and $\Pr[\mathrm{m} = 1 \wedge \mathrm{c} = 2] = 1/6$. Hence,

$$\Pr[\mathrm{m} = 1 \mid \mathrm{c} = 2] = \frac{1/6}{1/4} = \frac{2}{3} \neq \frac{1}{3} = \Pr[\mathrm{m} = 1].$$

Perfect secrecy lost

The probability that $m = 1$ given $c = 2$ is double what it was.

Once Eve sees $c = 2$ there are only two possibilities for m:

1. $m = 1$ (and $k = 1$)
2. $m = 2$ (and $k = 0$).

No longer possible that $m = 0$!

Eve narrows the possibilities for m to the set $M = \{1, 2\} \subseteq \mathcal{M}$. Her probabilistic knowledge of m changes from the initial distribution $(1/2, 1/3, 1/6)$ to the new distribution $(0, 2/3, 1/3)$. She has learned at lot about m, even without finding it exactly.

A seemingly minor change turns a cryptosystem with perfect secrecy into one that leaks a considerable amount of information!

Caveats with perfect secrecy

Perfect secrecy seems like the gold standard of security.

Nevertheless, even a scheme with perfect secrecy is not without defects and must still be used carefully.

Two problems:

- It succumbs immediately to a known plaintext attack.
- It is subject to a modification attack.

Known plaintext attack against simplified Caesar cipher

Suppose one knows even a single plaintext-ciphertext pair (m_1, c_1).

One easily solves the equation

$$c_1 = E_k(m_1) = (m_1 + k) \bmod 3$$

to find the key $k = (c_1 - m_1) \bmod 3$.

Hence, the system is completely broken.

Man-in-the-middle attacks

An *active attacker* is one who can both read and alter messages en route to their destinations.

We refer to such an attacker as "Mallory", and we call such an attack a *man-in-the-middle* attack.

In a *modification attack*, mallory can modify the contents of a message in specific semantically-meaningful ways even though he has no idea what the message actually is.

Modification attack against base Caesar cipher

Suppose Alice sends c to Bob. Mallory intercepts it and changes c to $(c + 5)$ mod 26.

Even though he doesn't know the key and cannot read m, he knows that he has changed m to $(m + 5)$ mod 26.

Why? Let's do the calculations. (All arithmetic is modulo 26).

$$D_k(c') = D_k(c + 5) = c + 5 - k = D_k(c) + 5 = m + 5.$$

Depending on the application, this could be a devastating attack. Suppose Alice were a financial institution that was making a direct deposit of m thousand dollars to Mallory's bank account at the Bob bank. By this attack, Mallory could get an extra 5 thousand dollars put into his account each month.

A modification attack on English vowels

In our encoding scheme, vowels are represented by even numbers: $A = 0$, $E = 4$, $I = 8$, $O = 14$, and $U = 20$. If m is a vowel, then $m' = (m + 5) \bmod 26$ is guaranteed not to be a vowel.

How could Mallory use this to his advantage?

A general's orders

- Suppose Alice is a general sending an order to a field commander whether or not to attack.
- She uses the Caesar cipher to encrypt the order.
- A vowel means to attack; a consonent to hold the position.
- She feels very clever for encoding the attack bit in such a non-obvious way.
- However, Mallory's $c + 5$ transformation changes every "attack" message to "don't attack" (and some "don't attack messages to "attack").
- This effectively prevents Alice from attacking when it is to her advantage.

The fact that she was using a cryptosystem for which perfect secrecy is known did not protect her.

Moral

The security of a system in practice depends critically on the kinds of attacks available to an attacker.

In this case, the cryptosystem that is provably perfectly secure against a passive eavesdropper using a ciphertext-only attack fails miserably against a known plaintext attack or against an active attacker.

Classical ciphers

One-time pad

The *one-time pad* is an information-theoretically secure cryptosystem that works for messages of arbitrary length.

It is important because

- it is sometimes used in practice;
- it is the basis for many stream ciphers, where the truly random key is replaced by a pseudo-random bit string.

It is based on *exclusive-or* (XOR), which we write as $\oplus$.

$x \oplus y$ is true when exactly one of x and y is true.
$x \oplus y$ is false when x and y are both true or both false.

Exclusive-or is just sum modulo two if 1 represents true and 0 represents false.

$$x \oplus y = (x + y) \bmod 2.$$

The one-time pad cryptosystem

$\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^r$ for some length r.

$E_k(m) = D_k(m) = k \oplus m$, where $\oplus$ is applied to corresponding bits of k and m.

XOR is associative and is its own inverse. Thus,

$$D_k(E_k(m)) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0 \oplus m = m.$$

Like the base Caesar cipher, for given m and c, there is exactly one key k such that $E_k(m) = c$ (namely, $k = m \oplus c$).

For fixed c, m varies over all possible messages as k ranges over all possible keys, so c gives no information about m.

It follows that the one-time pad is information-theoretically secure.

One-time pad in practice

The one-time pad would seem to be the perfect cryptosystem.

- It works for messages of any length (by choosing a key of the same length).
- It is easy to encrypt and decrypt.
- It is information-theoretically secure.

In fact, it is sometimes used for highly sensitive data.

It has two major drawbacks:

1. The key k must be as long as the message to be encrypted.
2. The same key must never be used more than once. (Hence the term "one-time".)

Together, these make the problem of key distribution and key management very difficult.

One-time pad vulnerable to a known plaintext attack

If Eve knows just one plaintext-ciphertext pair (m_1, c_1), then she can recover the key $k = m_1 \oplus c_1$.
This allows her to decrypt all future messages sent with that key.

Even in a ciphertext-only situation, if Eve has two ciphertexts c_1 and c_2 encrypted by the same key k, she can gain significant partial information about the corresponding messages m_1 and m_2.

In particular, she can compute $m_1 \oplus m_2$ without knowing either m_1 or m_2 since

$$m_1 \oplus m_2 = (c_1 \oplus k) \oplus (c_2 \oplus k) = c_1 \oplus c_2.$$

That information, together with other information she might have about the likely content of the messages, may be enough for her to seriously compromise the secrecy of the data.

Affine ciphers

Affine ciphers generalize simple shift ciphers such as Caesar.

Let α and β be two integers with $\gcd(\alpha, 26) = 1$.

A key is a pair $k = (\alpha, \beta)$.
There are 12 possible choices for α (1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25) and 26 possibilites for β, so $|\mathcal{K}| = 12 \times 26 = 312$.

Encryption: $E_k(m) = \alpha m + \beta \bmod 26$.

Decryption: $D_k(c) = \alpha^{-1}(c - \beta) \bmod 26$.

Here, α^{-1} is the multiplicative inverse of α in the ring of integers $\mathbf{Z}_{26}$. For example, $5^{-1} = 21$ since $21 \times 5 = 105 \equiv 1 \pmod{26}$.

α^{-1} exists precisely when $\gcd(\alpha, 26) = 1$.

Polyalphabetic ciphers

Another way to strengthen monoalphabetic ciphers is to use different substitutions for different letter positions.

- Choose r different alphabet permutations $k_1, \ldots, k_r$ for some number r.
- Use k_1 for the first letter of m, k_2 for the second letter, etc.
- Repeat this sequence after every r letters.

While this is much harder to break than monoalphabetic ciphers, letter frequency analysis can still be used.

Every r^{th} letter is encrypted using the same permutation, so the submessage consisting of just those letters still exhibits normal English language letter frequencies.

Vigenère cipher

The *Vigenère cipher* is a polyalphabetic cipher in which the number of different substitutions r is also part of the key. Thus, the adversary must determine r as well as discover the different substitutions.

All polyalphabetic ciphers can be broken using letter frequency analysis, but they are secure enough against manual attacks to have been used at various times in the past.

The German Enigma encryption machine used in the second world war is also based on a polyalphabetic cipher.

Hill cipher

A *polygraphic cipher* encrypts several letters at a time.
It tends to mask the letter frequencies, making it much harder to break.

The Hill cipher is such an example based on linear algebra.

- The key is, say, a non-singular 3×3 matrix K.
- The message m is divided into vectors m_i of 3 letters each.
- Encryption is just the matrix-vector product $c_i = Km_i$.
- Decryption uses the matrix inverse, $m_i = K^{-1}c_i$.

Unfortunately, the Hill cipher succumbs to a known plaintext attack. Given three linearly independent vectors m_1, m_2, and m_3 and the corresponding ciphertexts $c_i = Km_i$, $i = 1, 2, 3$, it is straightforward to solve for K.

Playfair cipher

The *Playfair* cipher, invented by Charles Wheatstone in 1854 but popularized by Lord Lyon Playfair, is another example of a polygraphic cipher [MvOV96, chapter 7, pp. 239-240] and [Wik].

Here, the key is a passphrase from which one constructs a 5×5 matrix of letters. Pairs of plaintext letters are then located in the matrix and used to produce a corresponding pair of ciphertext letters.

How Playfair works

Construct the matrix from the passphrase.

- Construct the matrix by writing the passphrase into the matrix cells from left to right and top to bottom.
- Omit any letters that have previously been used.
- Fill remaining cells with the letters of the alphabet that do not occur in the passphrase, in alphabetical order.
- In carrying out this process, "I" and "J" are identified, so we are effectively working over a 25-character alphabet.

Thus, each letter of the 25-character alphabet occurs exactly once in the resulting matrix.

Example Playfair matrix

Let the passphrase be

"CRYPTOGRAPHY REQUIRES STRONG KEYS".

The resulting matrix is

C	R	Y	P	T
O	G	A	H	E
Q	U	I/J	S	N
K	B	D	F	L
M	V	W	X	Z

First occurrence of each letter in the passphrase shown in orange:

"CRYPTOGRAPHY REQUIRES STRONG KEYS".

Letters not occurring in the passphrase: BDFLMVWXZ.

Encrypting in Playfair: preparing the message

To encrypt a message using Playfair:

- Construct the matrix.
- Remove spaces and pad the message with a trailing 'X', if necessary, to make the length even.
- Break up the message into pairs of letters.
- In case a pair of identical letters is about to be produced, insert an "X" to prevent that.

Examples:

- "MEET ME AT THE SUBWAY" becomes "ME" "ET" "ME" "AT" "TH" "ES" "UB" "WA" "YX".
- "A GOOD BOOK" becomes "AG", "OX", "OD" "BO", "OK".

Encrypting in Playfair: substituting the pairs

To encrypt pair *ab*, look at rectangle with *a* and *b* at its corners.

1. If *a* and *b* appear in different rows and different columns, replace each by the letter at the opposite end of the corresponding row. Example: replace "AT" by "EY":

   Y **T** P
   A E H

2. If *a* and *b* appear in the same row, then replace *a* by the next letter circularly to its right in the row, and similarly for *b*. For example, the encryption of "LK" is "KB".
3. If *a* and *b* appear in the same column, then replace *a* by the next letter circularly down in the column, and similarly for *b*.

Example: "MEET ME AT THE SUBWAY" encrypts as "ZONEZOEYPEHNBVYIPW".

Decrypting in Playfair

Decryption is by a similar procedure.

In decrypting, one must manually remove the spurious occurrences of "X" and resolve the "I/J" ambiguities.

See Trappe and Washington [TW06] for a discussion of how the system was successfully attacked by French cryptanalyst Georges Painvin and the Bureau du Chiffre.

Block ciphers

Block ciphers

A *block cipher* is an encryption system where the base message space $\mathcal{M}_0$ is finite. Elements of $\mathcal{M}_0$ are called *blocks*.
Blocks are typically bit strings of some convenient length such as 64 or 128.

A block cipher can be used in *electronic codebook (ECB) mode* to encrypt arbitrarily long messages:

1. Represent message m as a sequence of blocks $b_1, b_2, \ldots, b_r$.
2. Encrypt each block using the base cipher, so $c_i = E_k(b_i)$, $i \in [1 \ldots r]$. The same key k is used for each.
3. Output the sequence $c_1, c_2, \ldots, c_r$ of encrypted blocks.

Analysis of the Caesar cipher

The Caesar cipher is an example of a *block cipher*, where the blocks are single letters.

Although the Caesar cipher is not practical because of its small key space, many of its properties are representative of any block cipher used in ECB mode.

We now explore its security properties.

References


Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone.
Handbook of Applied Cryptography.
CRC Press, 1996.


Wade Trappe and Lawrence C. Washington.
Introduction to Cryptography with Coding Theory.
Prentice Hall, second edition, 2006.
ISBN 0-13-186239-1.


Wikipedia.
Playfair cipher.
URL `http://en.wikipedia.org/wiki/Playfair_cipher`.