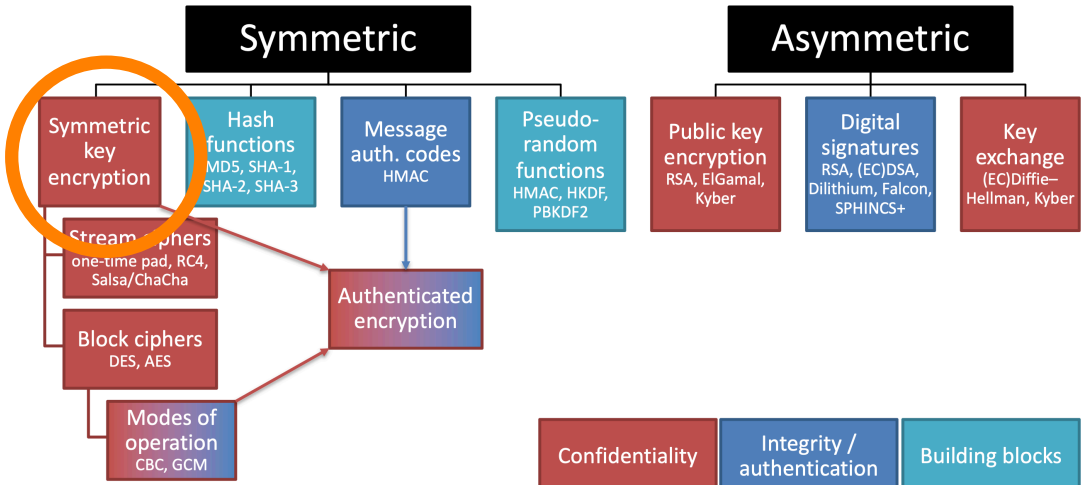
Topic 1.0
Symmetric encryption – Overview and historical ciphers

Douglas Stebila

CO 487/687: Applied Cryptography
Fall 2024

Map of cryptographic primitives



Outline

Caesar cipher and shift cipher

Defining symmetric-key encryption

Substitution cipher

Transposition cipher

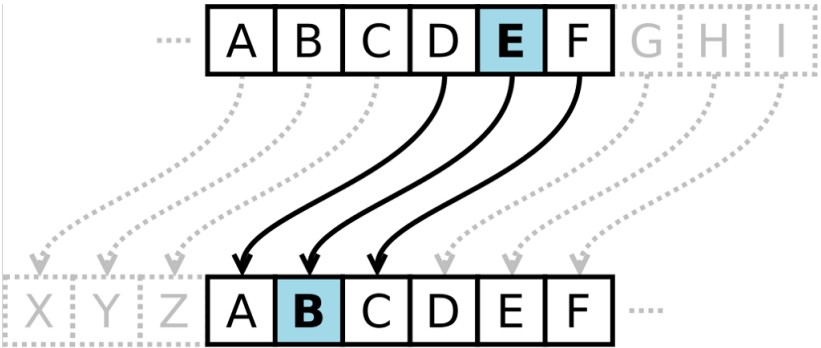
Defining security

Vigenère cipher

One-time pad a.k.a. Vernam cipher

Other historical ciphers

Caesar cipher



https://en.wikipedia.org/wiki/Caesar_cipher

Caesar cipher

ATTACK AT DAWN

↓

XQQXZH XQ AXTK

Caesar cipher

Encrypt(m) where $m \in \{A, \ldots, Z\}^*$

1 : **for** $i = 1, \ldots, |m|$
2 : $x \leftarrow \text{Encode}(m_i)$
3 : $y \leftarrow x + 23 \bmod 26$
4 : (or equivalently $y \leftarrow x - 3 \bmod 26$)
5 : $c_i \leftarrow \text{Decode}(y)$
6 : **return** c

Decrypt(c) where $c \in \{A, \ldots, Z\}^*$

1 : **for** $i = 1, \ldots, |c|$
2 : $x \leftarrow \text{Encode}(c_i)$
3 : $y \leftarrow x - 23 \bmod 26$
4 : (or equivalently $y \leftarrow x + 3 \bmod 26$)
5 : $m_i \leftarrow \text{Decode}(y)$
6 : **return** m

Encode / Decode:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Terminology

message/plaintext

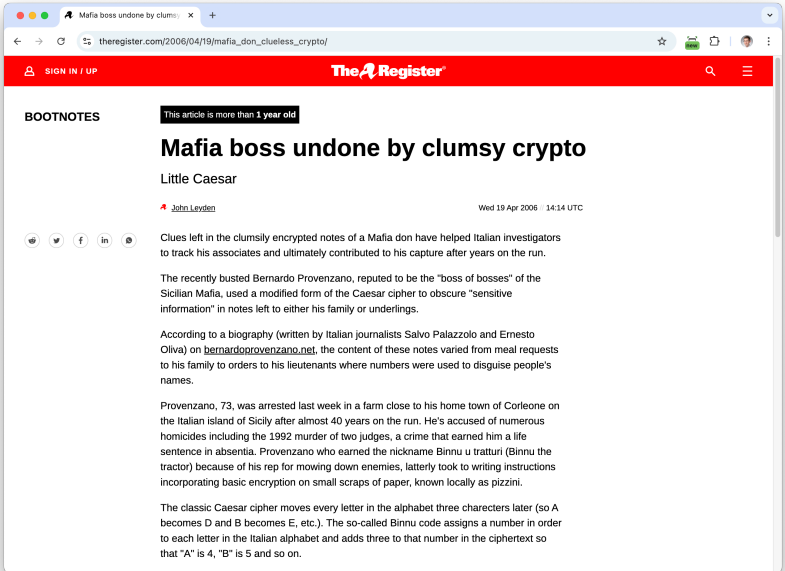
ciphertext

Note difference between “encrypt” and “encode”, and “decrypt” and “decode”:

- encode and decode map letters to numbers without trying to add security – just mapping to a more convenient space
- encrypt and decrypt try to add security

Is the Caesar cipher secure?

No.



https://www.theregister.com/2006/04/19/mafia_don_clueless_crypto/

Is the Caesar cipher secure?

No.

But also: what does it mean to be secure?

Principle 1.

Avoid security by obscurity.

Assume the adversary knows everything about the system–including how the algorithm works–except secret keys (“Kerckhoff’s principle”).

Shift cipher

Idea: modify Caesar cipher to obey Kerckhoff's principle by introducing a secret key.

Key space

$\mathcal{K} = \{0, \ldots, 25\}$

Randomly sample key $k \leftarrow_{\$} \mathcal{K}$

Encrypt(k, m) where $m \in \{A, \ldots, Z\}^*$

1 : **for** $i = 1, \ldots, |m|$
2 : $\quad x \leftarrow \mathsf{Encode}(m_i)$
3 : $\quad y \leftarrow x + k \bmod 26$
4 : $\quad c_i \leftarrow \mathsf{Decode}(y)$
5 : **return** c

Decrypt(k, c) where $c \in \{A, \ldots, Z\}^*$

1 : **for** $i = 1, \ldots, |c|$
2 : $\quad x \leftarrow \mathsf{Encode}(c_i)$
3 : $\quad y \leftarrow x - k \bmod 26$
4 : $\quad m_i \leftarrow \mathsf{Decode}(y)$
5 : **return** m

Is the shift cipher secure?

Breaking the shift cipher

Assumptions:

- We know the encryption/decryption algorithm being used but not any secret keys (Kerckhoff's principle)
- We are given a ciphertext to break ("ciphertext-only attack")

Can you recover the plaintext?

Two approaches:

1. Try all 26 possible secret key values $k = 0, \ldots, 25$. ("Brute-force attack" or "Exhaustive key search")
2. Frequency analysis.

Brute-force attack a.k.a. exhaustive key search

Given sufficient amounts of ciphertext c, decrypt c using each possible key k until c decrypts to a plaintext message which "makes sense".

How do we know when we've found the right key k that yields a plaintext that "makes sense"?

Have to make some assumption on the plaintext space:

- That it's a passage of English text:
  - In principle, 30 characters of ciphertext are sufficient on average to yield a unique plaintext that is a sensible English message. In practice, a few hundred characters are needed.
  - Need tools that can "recognize" sensible English.
- That it's a PDF file:
  - The first bytes of this PDF file are `%PDF-1.3`

Frequency analysis

Idea: compare the distribution of letters in the ciphertext with the distribution of letters in the underlying plaintext space.

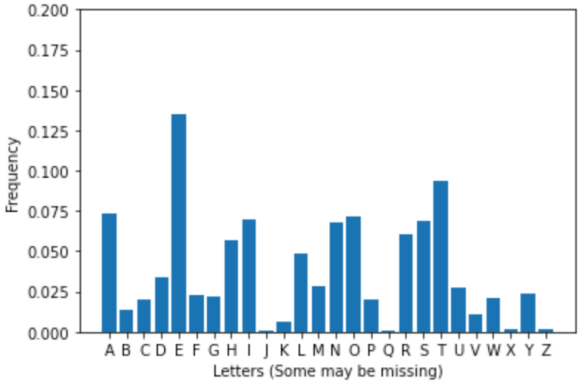
Additional assumption:

- The plaintext is a piece of English language text.

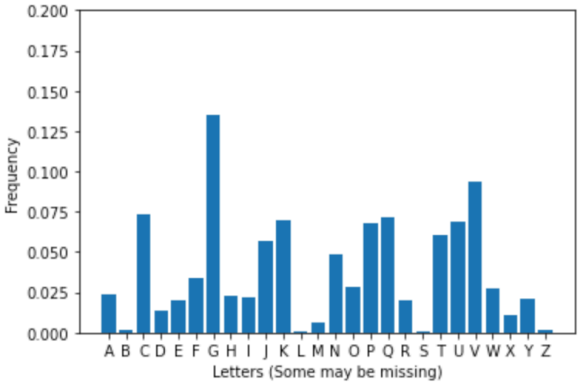
Key observation: Not all letters occur equally frequent in English text.

Frequency analysis of the shift cipher

Frequencies of letters in English text:



Frequencies of letters in a sample ciphertext from the shift cipher:



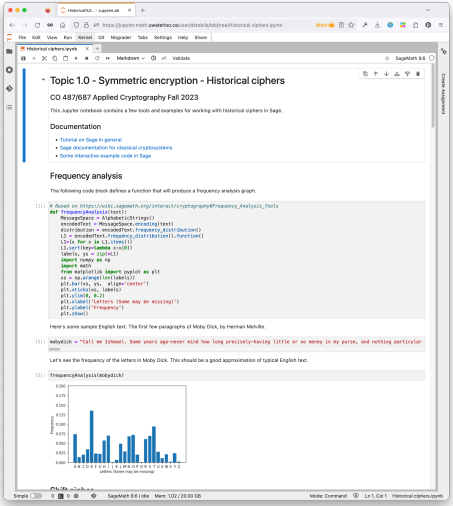
Based on the ciphertext letter frequencies, what is the most likely key? $k = 2$

Breaking the substitution cipher

The technique of frequency analysis was first described by Abu Yusuf Ya‘qub ibn Is-haq ibn as-Sabbah ibn ‘omran ibn Ismail al-Kindi, c. 820 A.D.

> *One way to solve an encrypted message, if we know its language, is to find a text of the same language long enough to fill one sheet or so and then we count each letter of it. We call the most frequently occurring letter the “first”, the next most occurring the “second”, the following most occurring the “third” and so on, until we finish all different letters in the cleartext. Then we look at the cryptogram we want to solve and we also classify its symbols. We find the most occurring symbol and change it to the form of the “first” letter, the next most common symbol is changed to the form of the “second” letter, and the following most common symbol is changed to the form of the “third” letter and so on, until we account for all the symbols of the cryptogram we want to solve.*

Interactive notebook for historical ciphers



Try out the interactive Jupyter notebook with Sage code implementing the historical ciphers from this class with frequency analysis graphs.

Download the code on LEARN → Content → Interactive notebooks, then run it on UW's Jupyter notebook server.

See instructions on LEARN.

Outline

Caesar cipher and shift cipher

Defining symmetric-key encryption

Substitution cipher

Transposition cipher

Defining security

Vigenère cipher

One-time pad a.k.a. Vernam cipher

Other historical ciphers

Defining symmetric-key encryption

Definition (Symmetric-key encryption scheme (SKES))

A symmetric-key encryption scheme (SKES) consists of:

- $\mathcal{M}$ – the plaintext space,
- $\mathcal{C}$ – the ciphertext space,
- $\mathcal{K}$ – the key space,
- an encryption algorithm, $E\colon \mathcal{K} \times \mathcal{M} \to \mathcal{C}$
- a decryption algorithm, $D\colon \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

such that $D(k, E(k, m)) = m$ for all $m \in \mathcal{M}$, $k \in \mathcal{K}$.

Equivalently: $E_k\colon \mathcal{M} \to \mathcal{C}$ and $D_k\colon \mathcal{C} \to \mathcal{M}$.

Outline

Caesar cipher and shift cipher

Defining symmetric-key encryption

Substitution cipher

Transposition cipher

Defining security

Vigenère cipher

One-time pad a.k.a. Vernam cipher

Other historical ciphers

The substitution cipher

Idea: instead of just shifting all letters by the same amount, have a table randomly mapping each letter to another fixed letter in a reversible way

Key space

$\mathcal{K} =$ all permutations of $\{A, \ldots, Z\}$

Message and ciphertext space

$\mathcal{M} = \mathcal{C} = \{A, \ldots, Z\}^*$

Encrypt(k, m)

1 : **for** $i = 1, \ldots, |m|$

2 : $\quad c_i \leftarrow$ apply permutation k to m_i

3 : **return** c

Decrypt(k, c)

1 : **for** $i = 1, \ldots, |c|$

2 : $\quad m_i \leftarrow$ apply inverse permutation k^{-1} to c_i

3 : **return** m

The substitution cipher

Example

$k =$

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	N	X	E	S	K	O	J	T	A	F	P	Y	I	Q	U	B	R	Z	G	V	C	H	M	W	L

$$m = \text{the big dog}$$

$$\begin{aligned} c &= E_k(\text{the big dog}) \\ &= \text{GJS NTO EQO} \end{aligned}$$

Is the substitution cipher secure?

Breaking the substitution cipher

Same assumptions as before:

- We know the encryption/decryption algorithm being used but not any secret keys (Kerckhoff's principle).
- We are given a ciphertext to break ("ciphertext-only attack").
- The plaintext is a piece of English language text.

Can you recover the plaintext?

Breaking the substitution cipher

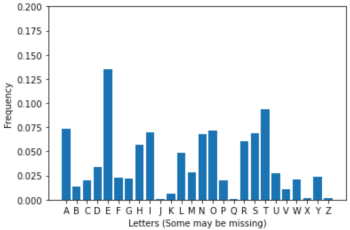
Will exhaustive key search work?

- Number of keys to try is $26! \approx 4 \times 10^{26} \approx 2^{88}$.
- If the adversary uses 10^6 computers, each capable of trying 10^9 keys per second, then exhaustive key search takes about 10^4 years.
- So, exhaustive key search is very expensive.

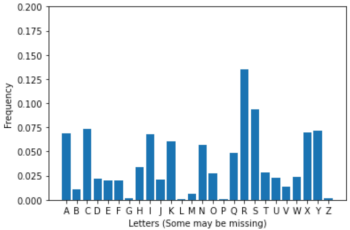
Will frequency analysis work?

Frequency analysis of the substitution cipher

Frequencies of letters in English text:



Frequencies of letters in a sample ciphertext from the substitution cipher:



Based on the ciphertext letter frequencies, what can you deduce?

Probably $E \to R$, $T \to S$

For this example, the key was:

`abcdefghijklmnopqrstuvwxyz`

`CVFHRUDNXLMQTIYEPKASOBJGWZ`

Frequency analysis of the substitution cipher

General method for frequency analysis:

1. Sort the characters in the ciphertext by frequency.
2. Guess that the ith most frequent ciphertext character represents the ith most frequent English letter.
3. Adjust the guess as needed until the message makes sense.

Can also do frequency analysis on two-letter combinations (“digrams”), three-letter combinations (“trigrams”), etc.

- e.g. `th` occurs more frequently than `zq`.

Outline

Caesar cipher and shift cipher

Defining symmetric-key encryption

Substitution cipher

Transposition cipher

Defining security

Vigenère cipher

One-time pad a.k.a. Vernam cipher

Other historical ciphers

The transposition cipher a.k.a. the permutation cipher

Idea: randomly rearrange the order of the letters

Fix a block length $B \in \mathbb{N}$.

Key space

$\mathcal{K} =$ all permutations of $\{1, \ldots, B\}$

Message and ciphertext space

$\mathcal{M} = \mathcal{C} = \cup_{i \geq 0} \{A, \ldots, Z\}^{iB}$

Encrypt(k, m)

1 : **for** $i = 1, \ldots, |m|/B$
2 : $x \leftarrow i$th block of m
3 : (in other words: $x \leftarrow m[(i-1)B \ldots iB - 1]$
4 : $y \leftarrow$ apply permutation k to positions of x
5 : ith block of $c \leftarrow y$
6 : **return** c

Decrypt(k, c)

1 : **for** $i = 1, \ldots, |c|/B$
2 : $y \leftarrow i$th block of c
3 : $x \leftarrow$ apply k^{-1} to positions of y
4 : ith block of $m \leftarrow x$
5 : **return** m

Transposition cipher

Example

Let $B = 5$ and $k = [41253]$ and encrypt the message $m =$"thisismymessage".

$k = [41253]$

$m =$

1	2	3	4	5
t	h	i	s	i
s	m	y	m	e
s	s	a	g	e

$c =$

4	1	2	5	3
S	T	H	I	I
M	S	M	E	Y
G	S	S	E	A

The ciphertext is $c =$"STHIIMSMEYGSSEA".

Is the transposition cipher secure?

Breaking the transposition cipher

Same assumptions as before:

- We know the encryption/decryption algorithm being used but not any secret keys (Kerckhoff's principle).
- We are given a ciphertext to break ("ciphertext-only attack").
- The plaintext is a piece of English language text.

Can you recover the plaintext?

- Exhaustive key search: Cost is $B!$. When B is large, this can be infeasible.
- Frequency analysis: Useless: single-letter frequency distribution stays the same since all we did is rearrange the letters.

Breaking the transposition cipher

Suppose we add another attack assumption:

- We are given the ability to have a plaintext of our choice encrypted under the secret key (“chosen plaintext attack”).

Now can you recover the plaintext?

Yes! Just ask for the plaintext “abcdefghijklmnopqrstuvwxyz” to be encrypted. Given the corresponding ciphertext, we can determine the permutation and thus the secret key.

Is it reasonable to have the additional assumption of chosen plaintext attack?

Outline

Caesar cipher and shift cipher

Defining symmetric-key encryption

Substitution cipher

Transposition cipher

Defining security

Vigenère cipher

One-time pad a.k.a. Vernam cipher

Other historical ciphers

What does it mean for a SKES to be secure?

Three key questions:

1. What is the adversary's goal?
2. How does the adversary interact with the communicating parties?
3. What are the computational powers of the adversary?

We always make the basic assumption of Kerckhoff's principle: The adversary knows everything about the SKES, except the particular key k chosen by Alice and Bob. (Avoid security by obscurity!!)

A security model or security definition contains three elements: 1) the adversary's goal, 2) how the adversary interacts with the communicating parties, and 3) the computational powers of the adversary.

Defining security: 1. Adversary's Goal

Some possible goals for the adversary in breaking symmetric key encryption:

1. Recover the secret key.
2. Systematically recover plaintext from ciphertext (without necessarily learning the secret key).
3. Learn some partial information about the plaintext from the ciphertext (other than its length).

If the adversary can achieve 1 or 2, the SKES is said to be totally insecure (or totally broken).

If the adversary cannot learn any partial information about the plaintext from the ciphertext (except possibly its length), the SKES is said to be semantically secure.

Hiding length information is very hard. This topic falls under the heading of *traffic analysis*.

Defining security: 2. Adversary's Interaction

Some possible ways the adversary might be able to interact with communicating parties:

- Passive attacks:
  - Ciphertext-only attack: The adversary only sees some encrypted ciphertexts.
  - Known-plaintext attack: The adversary also knows some plaintext and the corresponding ciphertext.
- Active attacks:
  - Chosen-plaintext attack: The adversary can also choose some plaintext(s) and obtain the corresponding ciphertext(s).
  - Chosen-ciphertext attack: The adversary can also choose some ciphertext(s) and obtain the corresponding plaintext(s). Includes the powers of chosen-plaintext attack.
- Other attacks:
  - Side-channel attacks: monitor the encryption and decryption equipment (timing attacks, power analysis attacks, electromagnetic-radiation analysis, etc.)
  - Physical attacks: bribery, blackmail, rubber hose, etc.

Defining security: 3. Computational Power of the Adversary

Some possible limits on the computational powers of the adversary:

- Information-theoretic security: The adversary has infinite computational resources.
- Complexity-theoretic security: The adversary is a "polynomial-time Turing machine".
- Computational security: The adversary has X number of real computers/workstations/supercomputers. ("computationally bounded"). Equivalently, the adversary can do X basic operations, e.g., CPU cycles.

Also have to consider:

- classical adversary: the adversary does not have access to a quantum computer
- quantum adversary: the adversary has access to a quantum computer

Work Factor

In this course:

- 2^{40} operations is considered very easy.
- 2^{56} operations is considered easy.
- 2^{64} operations is considered feasible.
- 2^{80} operations is considered barely feasible.
- 2^{128} operations is considered infeasible.

The Bitcoin network is presently performing hash operations at the rate of $654 \times 10^{18} \approx 2^{69.1}$ hashes per second (or $2^{94.1}$ per year), using 0.08% of the world's electricity consumption.

The Landauer limit from thermodynamics suggests that exhaustively trying 2^{128} symmetric keys would require $\gg 3000$ gigawatts of power for one year (which is $\gg$ 100% of the world's energy production).

https://ccaf.io/cbeci/index/comparisons

https://www.blockchain.com/explorer/charts/hash-rate

Security Level

Definition (Security level)

A cryptographic scheme is said to have a security level of ℓ bits if the fastest known attack on the scheme takes approximately 2^ℓ operations.

As of 2024, a security level of 128 bits is desirable in practice.

Defining semantic security as a “security game” or “security experiment”

To break semantic security under chosen plaintext attack, the adversary has to win the following game:

Security game: semantic security under chosen plaintext attack

1 : A secret key k is chosen at random.

2 : A challenge plaintext m^* is chosen, and we compute $c^* \leftarrow E_k(m^*)$.

3 : The adversary is given c^*.

4 : Chosen-plaintext attack: The adversary can select plaintexts m and gets the corresponding ciphertext $c = E_k(m)$.

5 : After a bounded amount of computation, the adversary outputs “some meaningful” information about m^*.

We measure the probability the adversary wins the game.

Indistinguishability-based definition of a Secure SKES

The game on the previous slide using semantic security is valid but hard to work with formally, so we typically deal with the following "indistinguishability"-based definition which can be proven to be equivalent:

Security game: indistinguishability under chosen plaintext attack (IND-CPA)

1 : A secret key k is chosen at random.

2 : Chosen-plaintext attack: The adversary can select plaintexts m and obtains the corresponding ciphertext $c = E_k(m)$.

3 : The adversary picks two messages m_0 and m_1 of the same length.

4 : The challenger picks random $b \xleftarrow{\$} \{0, 1\}$ and gives the adversary $c^* \leftarrow E_k(m_b)$.

5 : The adversary can again use their chosen-plaintext attack oracle.

6 : After a bounded amount of computation, the adversary outputs its guess b' for the hidden bit

We measure how much better than $1/2$ is the adversary's probability of winning the game.

Indistinguishability-based definition of a Secure SKES

Here is the IND-CPA security game defined in pseudocode, which is how you will normally see it in textbooks or papers. Let $\mathcal{A}$ be computationally bounded classical adversary.

Security game: indistinguishability under chosen plaintext attack (IND-CPA)

1 : $k \leftarrow_{\$} \mathcal{K}$ // $\leftarrow_{\$}$ means "sample uniformly at random from the given set"

2 : $(m_0, m_1) \leftarrow \mathcal{A}^{E_k(\cdot)}()$ // The superscript $E_k(\cdot)$ means the adversary can query an encryption oracle

3 : $b \leftarrow_{\$} \{0,1\}$ // and get the result, modelling a chosen-plaintext attack

4 : $c^* \leftarrow E_k(m_b)$

5 : $b' \leftarrow \mathcal{A}^{E_k(\cdot)}(c^*)$

6 : **if** $b' = b$ **then return** 1 **else return** 0

We measure the probability the experiment outputs 1, minus $1/2$, which is called the adversary's advantage: $\mathrm{Adv}^{\mathsf{INDCPA}}(\mathcal{A}) = \left|\mathrm{Prob}[\text{IND-CPA experiment outputs 1}] - \frac{1}{2}\right|$

Security against chosen ciphertext attacks

Let $\mathcal{A}$ be computationally bounded classical adversary.

Security game: indistinguishability under chosen ciphertext attack (IND-CCA)

1 : $k \leftarrow_\$ \mathcal{K}$

2 : $(m_0, m_1) \leftarrow \mathcal{A}^{E_k(\cdot), D_k(\cdot)}()$ // Adversary gets encryption and decryption oracles

3 : $b \leftarrow_\$ \{0,1\}$

4 : $c^* \leftarrow E_k(m_b)$

5 : $b' \leftarrow \mathcal{A}^{E_k(\cdot), D_k(\cdot \neq c^*)}(c^*)$ // But can't ask to decrypt the challenge ciphertext – that makes it trivial

6 : **if** $b' = b$ **then return** 1 **else return** 0

$$\mathrm{Adv}^{\mathrm{INDCCA}}(\mathcal{A}) = \left| \mathrm{Prob}[\text{IND-CCA experiment outputs 1}] - \frac{1}{2} \right|$$

Definition of a Secure SKES

We will generally want symmetric key encryption schemes to satisfy one of the following:

Definition (IND-CPA security)

Semantic security (equivalently, indistinguishability) under chosen-plaintext attack by a computationally bounded classical adversary.

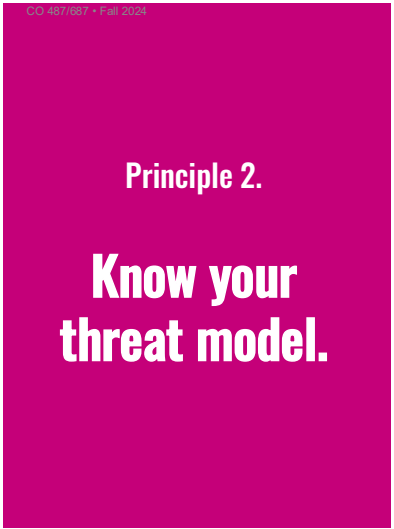
Definition (IND-CCA security)

Semantic security (equivalently, indistinguishability) under chosen-ciphertext attack by a computationally bounded classical adversary.

Elements of the definition:

1. Goal: semantic security (or equivalently, indistinguishability)
2. Adversary interaction: chosen-plaintext or chosen-ciphertext attack
3. Computational power: computationally bounded classical computer (or quantum if we can)

4+8+7 things to remember from CO 487



Security goal: confidentiality / integrity / authentication / …

How the adversary **interacts** with honest parties

Computational power of adversary: unlimited / asymptotic / bounded, classical / quantum

Plus physical security and more

Symmetric key primitives

Symmetric key encryption

Enc(k, m) → c
Dec(k, c) → m

- **Provides confidentiality.**
- Security goal: without secret key, infeasible to learn any information about plaintext (except possibly its length) even given the power to get arbitrary things encrypted/decrypted (“semantic security under chosen plaintext and ciphertext attack”).
- Secure options as of 2024:
  - block ciphers like AES in CBC mode
  - stream ciphers like ChaCha20 or AES in counter mode

Outline

Caesar cipher and shift cipher

Defining symmetric-key encryption

Substitution cipher

Transposition cipher

Defining security

Vigenère cipher

One-time pad a.k.a. Vernam cipher

Other historical ciphers

Vigenère cipher

Idea: Use different shift ciphers for different parts of the message to reduce effect of frequency analysis ("Polyalphabetic cipher")

Fix a block length $B \in \mathbb{N}$.

Key space

$\mathcal{K} = \{A, \ldots, Z\}^B$

Message and ciphertext space

$\mathcal{M} = \mathcal{C} = \cup_{i \geq 0} \{A, \ldots, Z\}^{iB}$

Encrypt(k, m)

1 : **for** $i = 1, \ldots, |m|$

2 : $\quad c_i \leftarrow m_i + k[i \bmod B] \bmod 26$

3 : **return** c

Decrypt(k, c)

1 : **for** $i = 1, \ldots, |c|$

2 : $\quad m_i \leftarrow c_i - k[i \bmod B] \bmod 26$

3 : **return** m

Vigenère cipher

Example with block length $B = 6$

$m =$	t	h	i	s		i	s		a	m	e	s	s	a	g	e
$+ k =$	C	R	Y	P		T	O		C	R	Y	P	T	O	C	R
$c =$	V	Y	G	H		B	G		C	D	C	H	L	O	I	V

Is the Vigenère cipher secure?

Security of the Vigenère cipher

Totally insecure against a known-plaintext attack. Why?

What about security under a ciphertext-only attack?

1. Determine the block length B (if not already known) using frequency analysis.
2. Determine the key using B separate frequency analyses.

Ciphertext-only attack on the Vigenère cipher

Determine the block length B as follows:

1. Make a guess for B
2. Check your guess as follows.
   - 2.1 Divide the ciphertext letters into B groups, $G_0, G_1, \ldots, G_{B-1}$, where the ith ciphertext letter is placed in group $G_{i \bmod B}$.
   - 2.2 Examine the frequency distributions of letters in each group.
   - 2.3 If each distribution "looks like" the expected distribution of letters from sensible English text, then the key length guess is probably correct.

Ciphertext-only attack on the Vigenère cipher

Once the key length B is determined, determine the key as follows:

1. The ciphertext letters in each group G_i were encrypted using a common shift cipher
2. Do shift cipher frequency analysis on each group G_i to determine k_i

Outline

Caesar cipher and shift cipher

Defining symmetric-key encryption

Substitution cipher

Transposition cipher

Defining security

Vigenère cipher

One-time pad a.k.a. Vernam cipher

Other historical ciphers

One-time pad a.k.a. Vernam cipher

Idea: A modification of the Vigenère cipher where the key is as long as the message.

Invented by Vernam in 1917 for the telegraph system.

Fix a message length $\ell \in \mathbb{N}$.

Key space

$\mathcal{K} = \{A, \ldots, Z\}^\ell$

Message and ciphertext space

$\mathcal{M} = \mathcal{C} = \{A, \ldots, Z\}^\ell$

Encrypt(k, m)

1 : **for** $i = 1, \ldots, \ell$

2 : $\quad c_i \leftarrow m_i + k_i \bmod 26$

3 : **return** c

Decrypt(k, c)

1 : **for** $i = 1, \ldots, \ell$

2 : $\quad m_i \leftarrow c_i - k_i \bmod 26$

3 : **return** m

One-time pad a.k.a. Vernam cipher

Example

$m =$	t	h	i	s	i	s	a	m	e	s	s	a	g	e
$+ k =$	Z	F	K	W	O	G	P	S	M	F	J	D	L	G
$c =$	S	M	S	P	W	Y	P	F	Q	X	C	D	R	K

Note: the key is as long as the message.

One-time pad for binary messages

Fix a message length $\ell \in \mathbb{N}$.

Key space

$\mathcal{K} = \{0,1\}^\ell$

Message and ciphertext space

$\mathcal{M} = \mathcal{C} = \{0,1\}^\ell$

Encrypt(k, m)

1 : **for** $i = 1, \ldots, \ell$
2 : $\quad c_i \leftarrow m_i \oplus k_i$
3 : **return** c

Decrypt(k, c)

1 : **for** $i = 1, \ldots, \ell$
2 : $\quad m_i \leftarrow c_i \oplus k_i$
3 : **return** m

Note that we can use exclusive-OR ($\oplus$) in both encryption and decryption since addition and subtraction modulo 2 are equivalent.

Re-use of one-time pads is bad

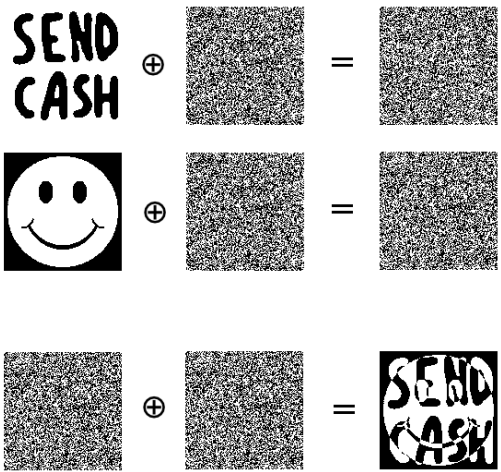
The key for a one-time pad cipher must not be re-used:

- If $c_1 = m_1 + k$ and $c_2 = m_2 + k$, then $c_1 - c_2 = m_1 - m_2$.
- $c_1 - c_2$ depends only on the plaintext (and not on the key) and hence can leak information about the plaintext.
- If m_1 is known, then m_2 can be easily computed.

Re-use of one-time pads is bad

Interactive demo: `https://www.douglas.stebila.ca/teaching/visual-one-time-pad/`

Re-use of one-time pads is bad



http://www.cryptosmith.com/archives/70

Security of the one-time pad

- Perfect secrecy: The one-time pad is semantically secure against ciphertext-only attack by an adversary with infinite computational resources.
- This can be proven formally using concepts from information theory [Shannon 1949].
- The bad news: Shannon also proved that if plaintexts are m-bit strings, then any symmetric-key encryption scheme with perfect secrecy must have $|K| \geq 2^m$.
- Perfect secrecy (and the one-time pad) is fairly useless in practice.
  - A re-used key can be broken.
  - A non-random key can (in principle) be broken.
  - Hard to safely distribute and manage truly random one-time keys.

Outline

Caesar cipher and shift cipher

Defining symmetric-key encryption

Substitution cipher

Transposition cipher

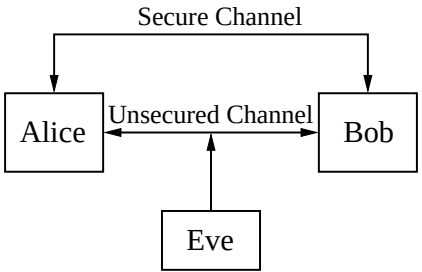
Defining security

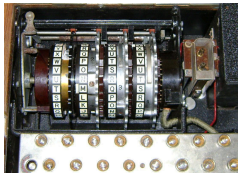
Vigenère cipher

One-time pad a.k.a. Vernam cipher

Other historical ciphers

World War II: Enigma Machine

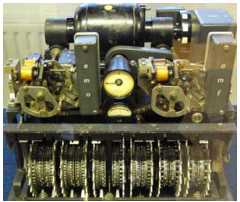


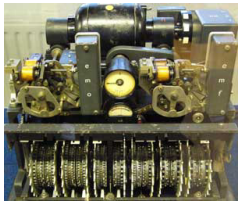


Alan Turing

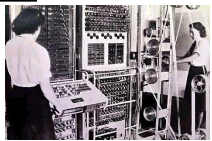
See https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma

World War II: Lorenz Machine





Bill Tutte



Colossus

See `http://tinyurl.com/COBillTutte`, `http://billtuttememorial.org.uk/`

Code Talkers

"A code talker was a person employed by the military during wartime to use a little-known language as a means of secret communication."

Used by the United States in WWI, WWII, and even up through the Korean War and early into the Vietnam War. Estimates are that 400-500 Native Americans were employed by the US Marine Corps.

[Comanche code talkers of the 4th Signal company during World War I]

https://en.wikipedia.org/wiki/Code_talker

Navajo Code Talkers in WWII

Most well-documented story is that of Navajo code talkers in WWII.

- Navajo has a complex grammar
- Not decipherable even by speakers of closely related languages
- Unwritten language at the time
- Numerous dialects
- At start of WWII, estimated fewer than 30 non-Navajo could understand the language
- Mostly used by the US in the Pacific theatre of war

[Navajo code talkers in the Pacific theatre, June 1944]

https://en.wikipedia.org/wiki/Code_talker

Navajo as a code

Type one code: Use Navajo words to represent English letters. (Substitution cipher on the alphabet.) Very slow.

Type two code: Translate English text into Navajo. (Substitution cipher, but much more complex.) Develop codebook of Navajo words/phrases to represent technical terms. (E.g. “shark” $\implies$ “destroyer”; “hummingbird” $\implies$ “fighter plane”.)

To convince the generals, a combat scenario was simulated with code talkers transmitting a 3-line message in 20 seconds, versus 30 minutes on code-wheel encryption machines.

Special dialect developed for military purposes, undecipherable even for native Navajo speakers.

“Were it not for the Navajos, the Marines would never have taken Iwo Jima.” – Major Howard Connor, 5th Marine Division signal officer

https://en.wikipedia.org/wiki/Code_talker, https://web.archive.org/web/20100327055830/https://www.cia.gov/news-information/featured-story-archive/2008-featured-story-archive/navajo-code-talkers/index.html

Canadian Cree Code Talkers

- Thousands of Indigenous people volunteered for service for Canada in WWII, many funnelled into the Canadian Army and sent to England.
- Some Cree, Ojibwe, Inuit, Métis, and others were then sent to London to test the idea of having code talkers in the European theatre of war.
- Cree speakers were then trained, selected, and deployed as code talkers with US armed forces.

https://www.thecanadianencyclopedia.ca/en/article/cree-code-talkers

Canadian Cree Code Talkers

- Sworn to secrecy after the war and not much information disclosed about Canadian Cree code talkers in the following decades.
- Came to light due to 2003 interview with Charles Tomkins and a 2017 documentary called *Cree Code Talker*.

https://www.youtube.com/watch?v=7JiUPBKST5M

https://www.cbc.ca/news/canada/edmonton/cree-language-used-as-secret-weapon-in-wwii-1.3150324