Lecture 1: Encryption

(First step towards security)

1.1: Definitions (basic): Encryption/decryption/keys

1.2: Classical ciphers + illustration of attacks

1.3: Definitions & properties of cryptosystems (more formal)

1.4: Modern ciphers: Stream Cipher

1.5: Modern ciphers: Block Ciphers + recommended key length

1.6: Attacks on cryptosystem implementations

1.7: Kerckhoffs’ principle vs security through obscurity

1.8: Interesting historical facts

How Important is Encryption?

Hadi Partovi, co-founder of Code.org:

"Encryption is at least as foundational as photosynthesis"

"We don't teach biology or chemistry to kids because they're going to become surgeons or chemists.

We teach them about photosynthesis and that water is H2O, or how lightbulbs work, just to understand the world around us.

You don't use any of it, but you **do on a day-to-day basis use *public-key encryption***"

Increasing Data Protection Need

The Economist | Topics | Current edition | More | Subscribe

Regulating the internet giants

The world's most valuable resource is no longer oil, but data

The data economy demands a new approach to antitrust rules

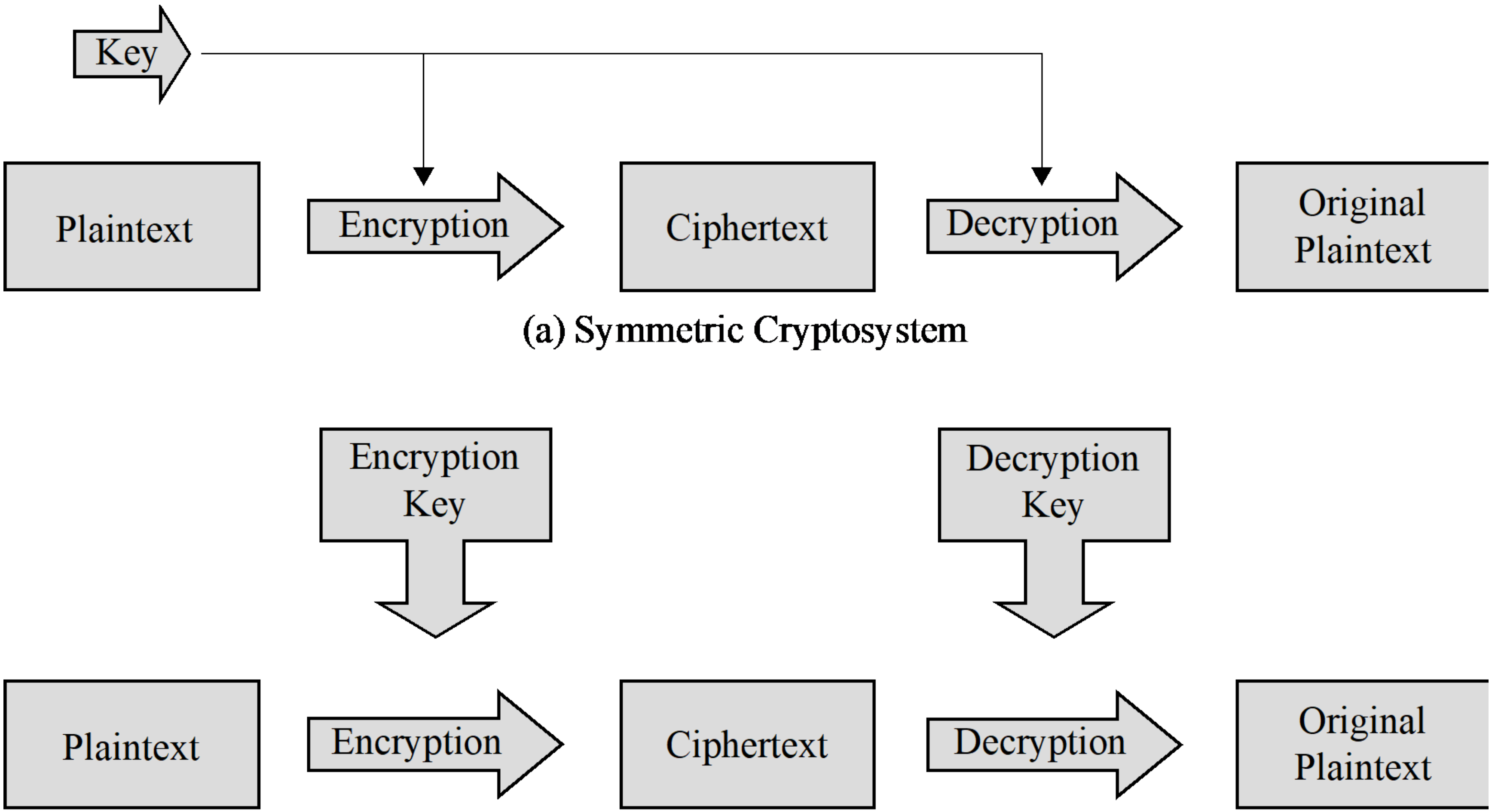
Print edition | Leaders ›

May 6th 2017

Ref: https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data

1.1 Definitions (Basic)

Note: ***Symmetric*** vs. Asymmetric Cryptosystems

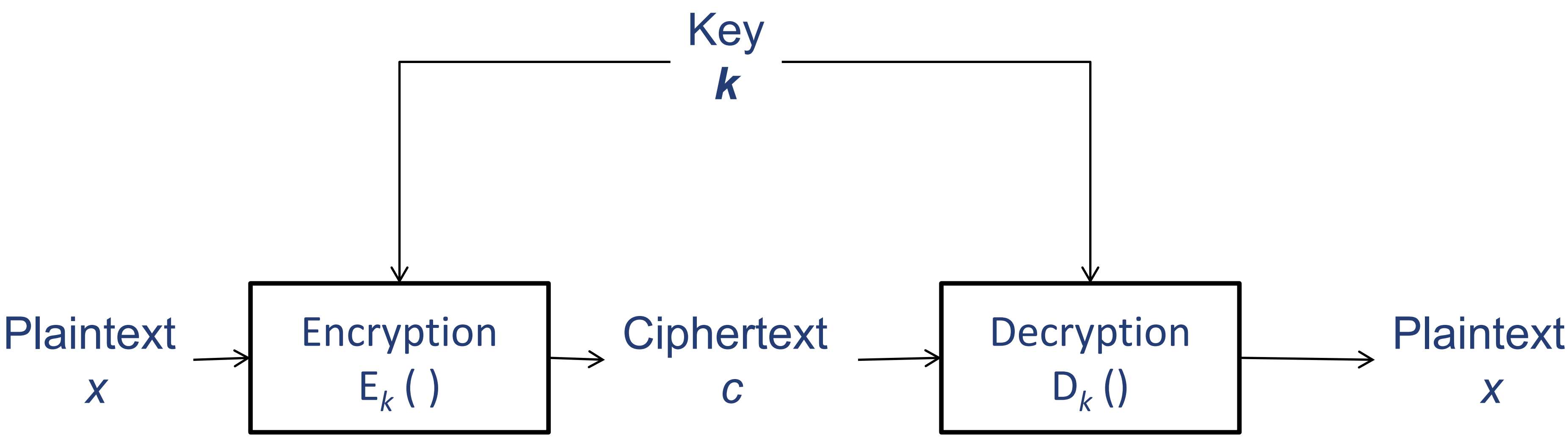


(a) Symmetric Cryptosystem

(b) Asymmetric Cryptosystem

Encryption

An **encryption scheme** (also known as **cipher**) consists of two algorithms:
encryption and decryption



Two requirements:

Correctness: For any plaintext x and key k, $D_k(E_k(x)) = x$

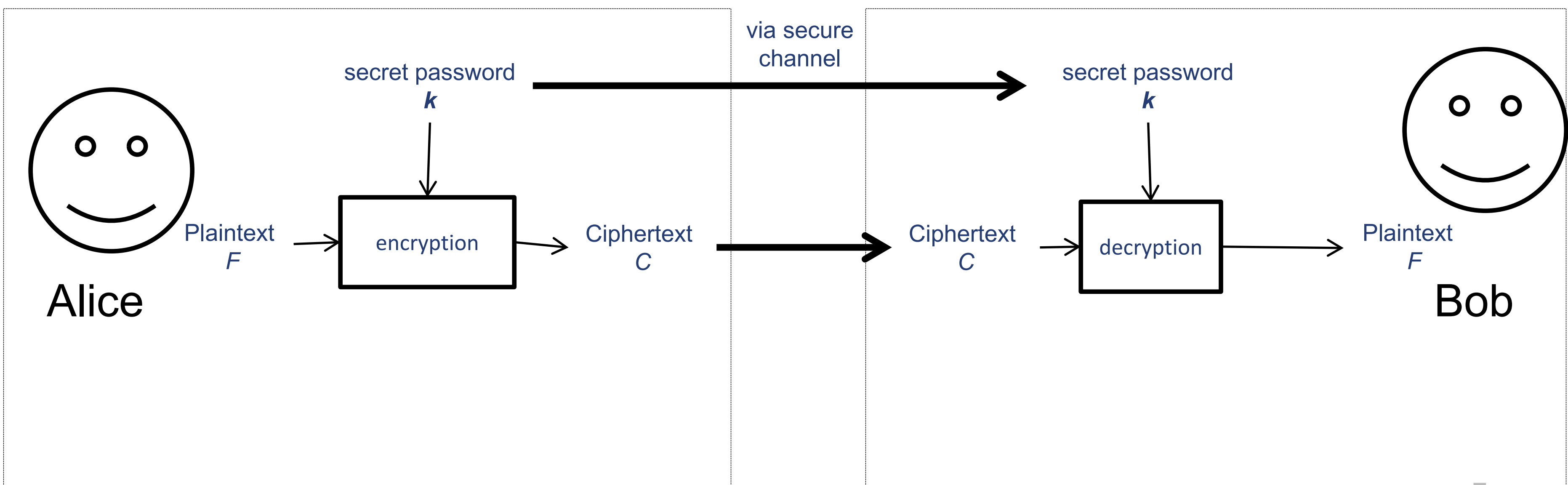
Security: Given the ciphertext, it is "difficult" to derive useful information of the key *k* and the plaintext *x*. The ciphertext should resemble a sequence of random bytes.

(There are many refined formulations of security requirements, e.g. *semantic security*.
In this module, we will not go into too much details, but some basic requirements are to be mentioned later.)

\+ ***Performance* requirement**: the encryption & decryption processes can be efficiently computed.

A Simple Application Scenario

- Alice had a large file ***F*** (say an **Excel file** containing information of her bank accounts and financial transactions).
- She "encrypted" the file ***F*** using **Winzip** with a password "13j8d7wjnd", and obtained the ciphertext ***C.***
- Next, she called Bob to tell him the password, and subsequently sent the ciphertext to Bob via email attachment.
- Later, Bob received ***C***, and decrypted the ciphertext with the password to recover the plaintext ***F***.



A Simple Application Scenario

- Anyone who has obtained ***C***, without knowing the password, is unable to get any information on ***F***
- Although *C* indeed contains information of *F*, the information is “hidden”
- To someone who doesn’t know the secret, *C* is just a **sequence of random bits**
- ***Remark***:
  Winzip is ***not*** an encryption scheme.
  It is an application that employs standard encryption schemes, such as AES.

Cryptography (vs Cryptology?)

- ***Cryptography*** is the study of techniques in **securing communication** in **the presence of adversaries** who have access to the communication
- Although cryptography is commonly associated with encryption, **encryption is *just one*** of the primitives in cryptography
- Others include cryptographic hash, digital signature, etc.
- *How about cryptology?*

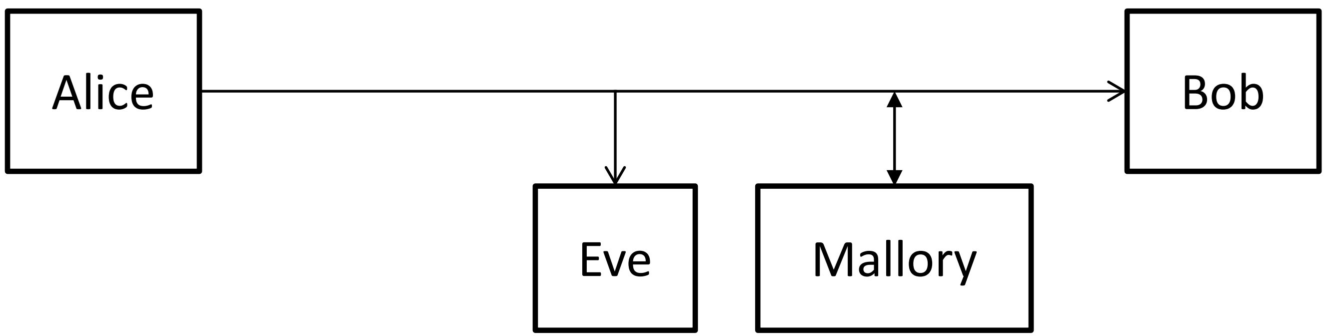
Cryptography

Cryptography is everywhere with **ubiquitous** applications now

- Secure communication:
  no message eavesdropping & tampering
- Secure transactions over the Internet
- Disk encryption: EFS, TrueCrypt/VeraCrypt
- Content protection (DRM)
- Passwords, password hashing
- Digital signatures: e.g. digitally signed software, documents
- Cryptocurrency: e.g. Bitcoin
- ...

Characters in Cryptography

- Common **placeholders** used in cryptography:
  - Alice: usually the originator of message
  - Bob: usually the recipient
  - Eve: eavesdropper, can only listen to sent messages
  - Mallory: malicious, can also modify sent messages



- See the interesting list of crypto characters in: https://en.wikipedia.org/wiki/Alice_and_Bob
- Depending on context, Alice may *not* be a human: she could be the machine that encrypts the message

In this module,
"read": Part of the teaching materials. Read it.
"see": Info that is good to know.
"optional": Optional information.

1.2 Classical Ciphers

For illustration, we will look into a few classical ciphers.
Classical ciphers are **not** secure in the computer era.
(The exception: the "unbreakable" one-time-pad).

(See http://ciphermachines.com/index
for a good listing of classical ciphers and cipher machines used during WWII.)

1.2.1. Substitution Cipher
1.2.2. Vigenere Cipher
1.2.3. Permutation Cipher
1.2.4. One Time Pad

1.2.1 Substitution Cipher

Substitution Cipher

- ***Plaintext*** and ***ciphertext***: a string over a set of symbols *U*

  E.g.

  Let *U*={“a”, “b”, “c”, …, “z”, “_”}.

  Example of plaintext: “hello_world”

- **Key**: a substitution table *S*,

  representing an 1-1 onto function from *U* to *U*

  E.g.

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	_
g	v	w	b	n	e	f	h	d	a	t	l	u	c	q	m	z	i	r	s	j	x	o	y	k	_	p

S(a) = g, S(b) = v, …

The inverse of S:

S^{-1}(g)=a, S^{-1} (v) = b

Substitution Cipher

Some terms:

- The ***key space***: the set of all possible keys
- The ***key space size***: the total number of possible keys
- The ***key size*** or ***key length***: the number of bits required to represent a particular key

- For substitution cipher:
  - The key space?
  - The key space size: 27!
  - The key size: **at least** $\log_2(27!) \approx 94$ bits

Showing the Lower Bound of the Key Size/Length

- The ***lower bound*** of **key size/length**: $\log_2(27!) \approx$ **94 bits**
- A few possible **key representations**:
  - **1 byte** per symbol/character: 27 * 1 byte = 27 bytes = **216** bits
  - **5 bits** per symbol/character: 27 * 5 bits = **135** bits
- How to show that 94 bits is the **lower bound**?
  - Show that using 94 bits is possible to represent all keys
  - Show that using <94 bits is not possible to represent all keys

- *So, how to show these??*
- To be answered next week

Substitution Cipher: Encryption/Decryption

Encryption: Given a plaintext of length n, which is a string $X = x_1\ x_2\ x_3\ \ldots\ x_n$, and the key S, output the ciphertext

$$E_S\ (X) = S(x_1)\ S(x_2)\ S(x_3)\ \ldots\ S(x_n)$$

Example:

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	_
g	v	w	b	n	e	f	h	d	a	t	l	u	c	q	m	z	i	r	s	j	x	o	y	k	_	p

plaintext: h e l l o _ w o r l d

ciphertext: h n l l q p o q i l b

Decryption: Given a string of ciphertext of length n $C = c_1\ c_2\ c_3\ \ldots\ c_n$ and the key S, output the plaintext

$$D_S(C) = S^{-1}(c_1)\ S^{-1}(c_2)\ S^{-1}(c_3)\ \ldots\ S^{-1}(c_n)$$

Attacks on a Cipher

- In general, the **attacker's goal** is:
  - To find **the key**: if the key can be found, then the plaintext can be obtained *(How about the converse?)*
  - To obtain some information of **the plaintext**

- Before commencing an attack, the attacker needs access to some information, such as:
  - A large number of ciphertexts that are all encrypted using the same key → **"ciphertext only"**; or
  - Pairs of ciphertext and the corresponding plaintext → **"known plaintext"**

Exhaustive Search (a.k.a. Brute-Force Search) Attack

- A ***simple (brute-force) attack*** is to exhaustively search the keys:
  - i.e. examine all possible keys one by one
- For most schemes, exhaustive search is *infeasible*
- Surprisingly, for some modern ciphers e.g. DES (key length of 56 bits), it is feasible to break it using exhaustive search
- More sophisticated attacks exploit the properties of the encryption scheme to spee dup the process

Exhaustive Search Attack: Known-Plaintext Scenario

- Consider the **substitution cipher** (with table size of 27)
- Suppose the attacker somehow has access to a **ciphertext *C* and a plaintext *X*,**
  how difficult for him to find the key using **exhaustive search**?
- Let ***S*** be the set of all possible substitution table
- Given *X*, *C*:

1. For each *S* in ***S***
2. Compute $X' = D_S(C)$;
3. If ($X' == X$) then break;
4. end-for
5. Display ("The key is ", *S*);

Exhaustive Search Attack: Known-Plaintext Scenario

- The **running time** depend on the size of the key space ***S***
- Since a key can be represented by a sequence of 27 symbols, the key space size is 27!

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	_
g	v	w	b	n	e	f	h	d	a	t	l	u	c	q	m	z	i	r	s	j	x	o	y	k	_	p

- Eventually, exhaustive search will find the key
- However, in the worst case, the exhaustive search needs to carry out **27! ≈ 2^{94} loops**, and on average ≈ 2^{93} loops.
  This is infeasible using current computing power (see Tutorial 1).
- *Can we attack substitution cipher **more efficiently**?*

Better Attack on Substitution Cipher: Known-Plaintext Attack

- "***Known-plaintext attack scenario***":
  when an adversary has access to pairs of ciphertexts and their corresponding plaintexts, and try to guess the key
- The attacker *doesn't* need to carry out exhaustive search. Given a plaintext and ciphertext, e.g.

  plaintext: `h e l l o _ w o r l d`

  ciphertext: `h n l l q p o q i l b`

  The attacker can figure out ***the entries in the key***

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	_
			b	n			h				l			q			i					o				p

- For a sufficiently long ciphertext, the **full table** can be determined

Better Attack on Substitution Cipher: Known-Plaintext Attack

- If the adversary can successfully derive the key, we say that the scheme is:

  “***insecure under known-plaintext attack***”

  or

  “***broken under known-plaintext attack***”

- Hence, **substitution cipher** is ***insecure*** under known-plaintext attack!

Some Remarks on Known-Plaintext Attack

- To carry out a known-plaintext attack, the attacker needs to obtain **at least a pair** of ciphertext and its corresponding plaintext
- *Is this requirement reasonable??*
- In many cases, the attacker *doesn't need* to know the full plaintext: only ***the first few bytes*** of the plaintext are sufficient
- These first few bytes of the plaintext can sometimes be **guessed**:
  - **Email data**: certain words in its header are fixed, such as "From", "Subject", etc.
  - Many **network protocols** have fixed headers,
    or only a few choices in their first few bytes of data packets
  - During WWII, cryptologists exploited **commonly-used words** like "weather" and "nothing to report" as the known plaintext to guess the secret keys
    (Optional: Read more about the "Enigma Machine".)

Exhaustive Search Attack: Ciphertext-Only Attack

- Suppose the attackers have access to **ciphertext only** (i.e. without the corresponding plaintext), and knows that the plaintext are **English sentences**.
- Can he successfully find the key using **exhaustive search**?
- *Yes!*
- Let ***S*** be the set of all possible substitution table
  Given *C*:

  1. For each *S* in ***S***
  2. Compute $X = D_S(C)$;
  3. If *X* contains **words in the English dictionary**, then break;
  4. end-for
  5. Display ("The key is ", *S*);

Exhaustive Search Attack: Ciphertext-Only Attack

- Eventually, the exhaustive search will **find the key**
- **Note**: There is a very small probability that the above algorithm **finds a wrong key**.
  Yet, for a sufficiently long ciphertext, e.g. 50 characters long, the probability that a wrong key will give a meaningful English sentence is ***very low*** (treated as "***negligible***").
- However, the attack is also *infeasible* due to the large key space size
- *Is there an **effective** ciphertext-only attack technique on substitution cipher?*
- Yes!

Frequency Analysis

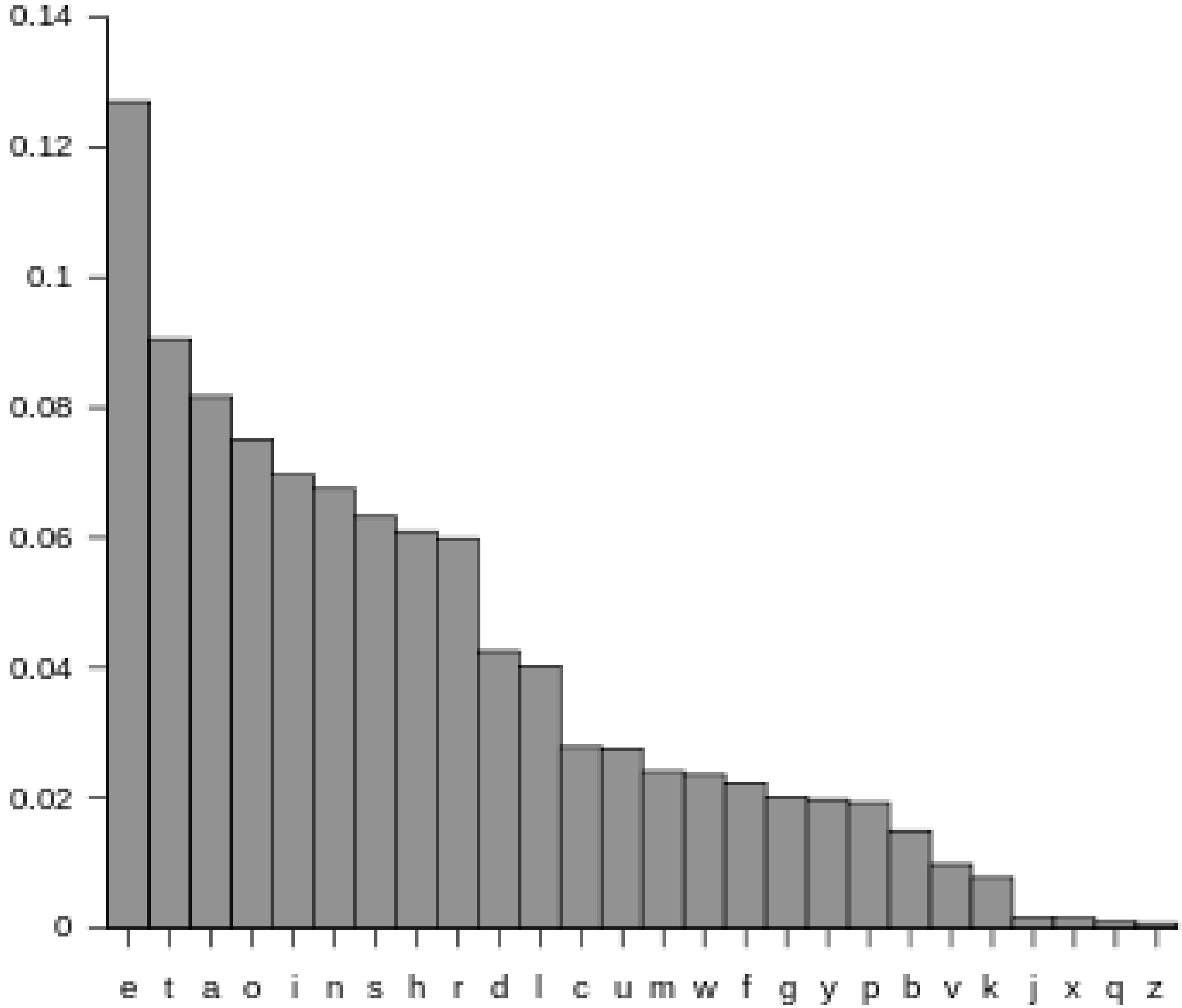
- Substitution cipher is vulnerable to ***frequency analysis***
- Note that in the hello_world example below,
symbol "o" appears 2 times in the plaintext, whereas the corresponding "q" also appears 2 times in the ciphertext

plaintext:	h	e	l	l	o	_	w	o	r	l	d
ciphertext:	h	n	l	l	q	p	o	q	i	l	b

- **A *monoalphabetic* cipher**: the substitution is fixed for *each letter* of the alphabet

- Suppose the plaintexts are English sentences.
The ***letter frequency distribution*** in English text is ***not*** uniform, for e.g. "e" is more commonly used than "z".
- *How can an adversary apply frequency analysis and break substitution cipher?*

Letter Frequency Distribution in English Text



From http://en.wikipedia.org/wiki/Letter_frequency

Frequency Analysis on Substitution Cipher

- If adversary knows that the plaintexts are English sentences, given a sufficiently long ciphertext (e.g. ≥ 50 characters), then an adversary may be able to guess the plaintext by:
  - **Mapping** the *frequently-occurring letters in the ciphertext*
    to
    the *frequently-occurring letters of English*
- Frequency analysis can be successfully carried out!
- Hence, substitution cipher is ***not secure under ciphertext-only attack*** either, when the plaintexts are English sentences
- In fact, the attack is effective on any human language

LumiNUS Forum Challenge: Just for a Good Fun

- "**Breaking substitution cipher**" challenge in LumiNUS forum
- You'll be given a ciphertext
- Do break the substitution cipher by finding the correct corresponding plaintext
- You can also refer to the uploaded **"Self-Exploration Activity 1"**

- The ***first person*** who can post the correct plaintext will get **3 (three) extra marks** for assignment (capped at 25)!
- The challenge will be posted **next week (20 Aug)** at **~9pm**