Part I

Secret-key cryptosystems basics

PROLOGUE - I.

Decrypt cryptotexts:

Decrypt cryptotexts:

GBLVMUB JOGPSNBUJLZ

Decrypt cryptotexts:

GBLVMUB JOGPSNBUJLZ

VMNIR

PROLOGUE - I.

Decrypt cryptotexts:

GBLVMUB JOGPSNBUJLZ

VMNIR

RPNBMZ EBMFLP OFABKEFT

PROLOGUE - II.

PROLOGUE - II.

Decrypt:

Decrypt:

VHFUHW GH GHXA
VHFUHW GH GLHX,
VHFUHW GH WURLV,
VHFUHW GH WRXV.

CHAPTER 4: SECRET-KEY (SYMMETRIC) CRYPTOGRAPHY

- In this chapter we deal with some of the very old, or quite old, classical (secret-key or symmetric) cryptosystems and their cryptanalysis that were primarily used in the pre-computer era.

CHAPTER 4: SECRET-KEY (SYMMETRIC) CRYPTOGRAPHY

- In this chapter we deal with some of the very old, or quite old, classical (secret-key or symmetric) cryptosystems and their cryptanalysis that were primarily used in the pre-computer era.
- These cryptosystems are too weak nowadays, too easy to break, especially with computers.

CHAPTER 4: SECRET-KEY (SYMMETRIC) CRYPTOGRAPHY

- In this chapter we deal with some of the very old, or quite old, classical (secret-key or symmetric) cryptosystems and their cryptanalysis that were primarily used in the pre-computer era.
- These cryptosystems are too weak nowadays, too easy to break, especially with computers.
- However, these simple cryptosystems give a good illustration of several of the important ideas of the cryptography and cryptanalysis.

CHAPTER 4: SECRET-KEY (SYMMETRIC) CRYPTOGRAPHY

- In this chapter we deal with some of the very old, or quite old, classical (secret-key or symmetric) cryptosystems and their cryptanalysis that were primarily used in the pre-computer era.
- These cryptosystems are too weak nowadays, too easy to break, especially with computers.
- However, these simple cryptosystems give a good illustration of several of the important ideas of the cryptography and cryptanalysis.
- Moreover, most of them can be very useful in combination with more modern cryptosystem - to add a new level of security.

BASICS

CRYPTOLOGY - HISTORY + APPLICATIONS

Cryptology (= cryptography + cryptanalysis)
has more than four thousand years long history.

CRYPTOLOGY - HISTORY + APPLICATIONS

Cryptology (= cryptography + cryptanalysis)
has more than four thousand years long history.

Some historical observation

CRYPTOLOGY - HISTORY + APPLICATIONS

Cryptology (= cryptography + cryptanalysis)
has more than four thousand years long history.

Some historical observation

- People have always had fascination with keeping information away from others.

CRYPTOLOGY - HISTORY + APPLICATIONS

Cryptology (= cryptography + cryptanalysis)
has more than four thousand years long history.

Some historical observation

- People have always had fascination with keeping information away from others.
- Some people – rulers, diplomats, military people, businessmen – have always had needs to keep some information away from others.

CRYPTOLOGY - HISTORY + APPLICATIONS

Cryptology (= cryptography + cryptanalysis)
has more than four thousand years long history.

Some historical observation

- People have always had fascination with keeping information away from others.
- Some people – rulers, diplomats, military people, businessmen – have always had needs to keep some information away from others.

Importance of cryptography nowadays

CRYPTOLOGY - HISTORY + APPLICATIONS

Cryptology (= cryptography + cryptanalysis)
has more than four thousand years long history.

Some historical observation

- People have always had fascination with keeping information away from others.
- Some people – rulers, diplomats, military people, businessmen – have always had needs to keep some information away from others.

Importance of cryptography nowadays

- Applications: cryptography is the key tool to make modern information transmission secure, and to create secure information society.

CRYPTOLOGY - HISTORY + APPLICATIONS

Cryptology (= cryptography + cryptanalysis)
has more than four thousand years long history.

Some historical observation

- People have always had fascination with keeping information away from others.
- Some people – rulers, diplomats, military people, businessmen – have always had needs to keep some information away from others.

Importance of cryptography nowadays

- Applications: cryptography is the key tool to make modern information transmission secure, and to create secure information society.
- Foundations: cryptography gave rise to several new key concepts of the foundation of informatics: one-way functions, computationally perfect pseudorandom generators, zero-knowledge proofs, holographic proofs, program self-testing and self-correcting, . . .

Classical cryptography used to concentrate till the end of the last millenium on designing and breaking encryption systems and all that in the context of classical information processing.

Modern cryptography has

Classical cryptography used to concentrate till the end of the last millenium on designing and breaking encryption systems and all that in the context of classical information processing.

Modern cryptography has

(1) significantly enlarged its scope to the rigorous analysis of any system that can be potential subject to malicious threats and to designs of such versions of such systems that can guarantee that they withstand such treats.

MAIN DEVELOPMENTS IN CRYPTOGRAOHY

Classical cryptography used to concentrate till the end of the last millenium on designing and breaking encryption systems and all that in the context of classical information processing.

Modern cryptography has

(1) significantly enlarged its scope to the rigorous analysis of any system that can be potential subject to malicious threats and to designs of such versions of such systems that can guarantee that they withstand such treats.

(2) started to develop cryptographic systems that also utilize elements and processes of the quantum world.

CONTINUATION

CONTINUATION

As one onsequence, many goals have been added to old cryptography to that of private and secure commuicaions in the presence of an adversary,

CONTINUATION

As one onsequence, many goals have been added to old cryptography to that of private and secure commuicaions in the presence of an adversary,

For example, digital signatures, authentication, privacy preservations, secret sharing, hashing, pseudorandom generators, zero-knowledge proofs, steganography, and so on.

CONTINUATION

As one onsequence, many goals have been added to old cryptography to that of private and secure commuicaions in the presence of an adversary,

For example, digital signatures, authentication, privacy preservations, secret sharing, hashing, pseudorandom generators, zero-knowledge proofs, steganography, and so on.

As another consequence, cryptography has moved from an engineering art, built on heuristic techniques, to a scientific disciplin based on mathematically rigorous design requirements, solution techniques and correctness proofs.

Such broadly developed modern cryptography is the subject of this lecture.

APPROACHES and PARADOXES in CRYPTOGRAPHY

Sound approaches to cryptography

Sound approaches to cryptography

- Shannon's approach based on **information theory**

Sound approaches to cryptography

- Shannon's approach based on **information theory** (**Enemy could not have enough information to break a given cryptosystem**).

Sound approaches to cryptography

- Shannon's approach based on **information theory** (**Enemy could not have enough information to break a given cryptosystem**).
- Current approach based on **complexity theory.**

Sound approaches to cryptography

- Shannon's approach based on **information theory** (**Enemy could not have enough information to break a given cryptosystem**).
- Current approach based on **complexity theory.** **(Enemy could not have enough computation power to break a given cryptosystem).**

Sound approaches to cryptography

- Shannon's approach based on **information theory** (**Enemy could not have enough information to break a given cryptosystem**).
- Current approach based on **complexity theory.** **(Enemy could not have enough computation power to break a given cryptosystem).**
- Very recent a new approach has been developed that is based on the laws and limitations of **quantum physics.**

Sound approaches to cryptography

- Shannon's approach based on **information theory** (**Enemy could not have enough information to break a given cryptosystem**).
- Current approach based on **complexity theory. (Enemy could not have enough computation power to break a given cryptosystem).**
- Very recent a new approach has been developed that is based on the laws and limitations of **quantum physics. (Enemy would need to break laws of nature in order to break a given cryptosystem).**

Sound approaches to cryptography

- Shannon's approach based on **information theory** (**Enemy could not have enough information to break a given cryptosystem**).
- Current approach based on **complexity theory. (Enemy could not have enough computation power to break a given cryptosystem).**
- Very recent a new approach has been developed that is based on the laws and limitations of **quantum physics. (Enemy would need to break laws of nature in order to break a given cryptosystem).**

Paradoxes of modern cryptography:

Sound approaches to cryptography

- Shannon's approach based on **information theory** (**Enemy could not have enough information to break a given cryptosystem**).
- Current approach based on **complexity theory.** **(Enemy could not have enough computation power to break a given cryptosystem).**
- Very recent a new approach has been developed that is based on the laws and limitations of **quantum physics.** **(Enemy would need to break laws of nature in order to break a given cryptosystem).**

Paradoxes of modern cryptography:

- **Positive results** of modern cryptography are based on **negative results** of computational complexity theory.

Sound approaches to cryptography

- Shannon's approach based on **information theory** (**Enemy could not have enough information to break a given cryptosystem**).
- Current approach based on **complexity theory.** (**Enemy could not have enough computation power to break a given cryptosystem**).
- Very recent a new approach has been developed that is based on the laws and limitations of **quantum physics.** (**Enemy would need to break laws of nature in order to break a given cryptosystem**).

Paradoxes of modern cryptography:

- **Positive results** of modern cryptography are based on **negative results** of computational complexity theory.
- Computers, that were designed originally for decryption, seem to be now more useful for **encryption**.

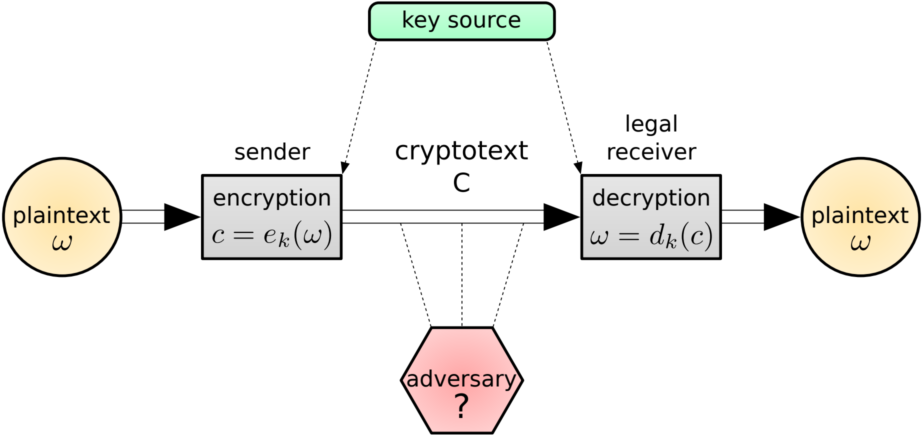
SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS - CIPHERS

The cryptography deals with problem of sending a message (plaintext, ciphertext, cleartext), through an insecure channel, that may be tapped by an adversary (eavesdropper, cryptanalyst), to a legal receiver.

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS - CIPHERS

The cryptography deals with problem of sending a message (plaintext, ciphertext, cleartext), through an insecure channel, that may be tapped by an adversary (eavesdropper, cryptanalyst), to a legal receiver.

Secret-key (symmetric) cryptosystems scheme:



SECRET-KEY (PRIVATE-KEY - SYMMETRIC) CRYPTOSYSTEMS

A secret-key (private-key or symmetric) cryptosystem is the one where the sender and the recepient share a common and secret key.

SECRET-KEY (PRIVATE-KEY - SYMMETRIC) CRYPTOSYSTEMS

A secret-key (private-key or symmetric) cryptosystem is the one where the sender and the recepient share a common and secret key.

Security of such a cryptosystem depends solely on the secrecy of shared key.

COMPONENTS of CRYPTOSYSTEMS:

Plaintext-space: P – a set of plaintexts (messages) over an alphabet $\sum$

Cryptotext-space: C – a set of cryptotexts (ciphertexts) over alphabet Δ

Key-space: K – a set of keys

COMPONENTS of CRYPTOSYSTEMS:

Plaintext-space: P – a set of plaintexts (messages) over an alphabet $\sum$

Cryptotext-space: C – a set of cryptotexts (ciphertexts) over alphabet Δ

Key-space: K – a set of keys

Each key $k \in K$ determines an **encryption algorithm e_k** and an **decryption algorithm d_k**

COMPONENTS of CRYPTOSYSTEMS:

Plaintext-space: P – a set of plaintexts (messages) over an alphabet $\sum$

Cryptotext-space: C – a set of cryptotexts (ciphertexts) over alphabet Δ

Key-space: K – a set of keys

Each key $k \in K$ determines an **encryption algorithm e_k** and an **decryption algorithm d_k** such that, for any plaintext $w, e_k(w)$ is the corresponding cryptotext and

$$w \in d_k(e_k(w)) \quad \text{or} \quad w = d_k(e_k(w)).$$

COMPONENTS of CRYPTOSYSTEMS:

Plaintext-space: P – a set of plaintexts (messages) over an alphabet $\sum$

Cryptotext-space: C – a set of cryptotexts (ciphertexts) over alphabet Δ

Key-space: K – a set of keys

Each key $k \in K$ determines an **encryption algorithm e_k** and an **decryption algorithm d_k** such that, for any plaintext $w, e_k(w)$ is the corresponding cryptotext and

$$w \in d_k(e_k(w)) \quad \text{or} \quad w = d_k(e_k(w)).$$

Note: As encryption algorithms we can use also **randomized algorithms.**

SECRET-KEY CRYPTOGRAPHY BASICS - SUMMARY

SECRET-KEY CRYPTOGRAPHY BASICS - SUMMARY

Symmetric cryptography relies on three algorithms:

Symmetric cryptography relies on three algorithms:

Key generating algorithm which generates a secret key in a cryptographically (pseudo)random way.

SECRET-KEY CRYPTOGRAPHY BASICS - SUMMARY

Symmetric cryptography relies on three algorithms:

Key generating algorithm which generates a secret key in a cryptographically (pseudo)random way.

Encryption algorithm which transforms a plaintext into a cryptotext using a secret key.

SECRET-KEY CRYPTOGRAPHY BASICS - SUMMARY

Symmetric cryptography relies on three algorithms:

Key generating algorithm which generates a secret key in a cryptographically (pseudo)random way.

Encryption algorithm which transforms a plaintext into a cryptotext using a secret key.

Decryption algorithm which transforms a cryptotext into the original plaintext using the same secret key.

SECRET-KEY CRYPTOGRAPHY BASICS - SUMMARY

Symmetric cryptography relies on three algorithms:

Key generating algorithm which generates a secret key in a cryptographically (pseudo)random way.

Encryption algorithm which transforms a plaintext into a cryptotext using a secret key.

Decryption algorithm which transforms a cryptotext into the original plaintext using the same secret key.

Secret key cryptosystems provide secure transmission of messages along insecure channel provided **the secret keys are transmitted over an extra secure channel.**

SECURITY of CRYPTOSYSTEMS

SECURITY of CRYPTOSYSTEMS

There are three fundamentally different ways a cryptosystem/cipher can be seen as secure.

SECURITY of CRYPTOSYSTEMS

There are three fundamentally different ways a cryptosystem/cipher can be seen as secure.

Unconditional security: is in the case it can be proven that the cryptosystem cannot be broken no matter how much power has the enemy (eavesdropper).

SECURITY of CRYPTOSYSTEMS

There are three fundamentally different ways a cryptosystem/cipher can be seen as secure.

Unconditional security: is in the case it can be proven that the cryptosystem cannot be broken no matter how much power has the enemy (eavesdropper).

Computational security is in the case it can be proven that no eavesdropper can break the cryptosystem in polynomial (reasonable) time..

SECURITY of CRYPTOSYSTEMS

There are three fundamentally different ways a cryptosystem/cipher can be seen as secure.

Unconditional security: is in the case it can be proven that the cryptosystem cannot be broken no matter how much power has the enemy (eavesdropper).

Computational security is in the case it can be proven that no eavesdropper can break the cryptosystem in polynomial (reasonable) time..

Practical security is in the case no one was able to break the cryptosystem so far after many years and many attempts.

WHO ARE CODEBREAKERS - DEVELOPMENTS

The vision of codebreakers has changed through the history, depending on the tools used for encryption and cryptoanalysis.

WHO ARE CODEBREAKERS - DEVELOPMENTS

The vision of codebreakers has changed through the history, depending on the tools used for encryption and cryptoanalysis.

- **Old times view:**

WHO ARE CODEBREAKERS - DEVELOPMENTS

The vision of codebreakers has changed through the history, depending on the tools used for encryption and cryptoanalysis.

- **Old times view:** Cryptology is a **black art** and crypanalysts communicate with **dark spirits** and even they are **followers of the devil**.

WHO ARE CODEBREAKERS - DEVELOPMENTS

The vision of codebreakers has changed through the history, depending on the tools used for encryption and cryptoanalysis.

- **Old times view:** Cryptology is a **black art** and crypanalysts communicate with **dark spirits** and even they are **followers of the devil**.
- **Pre-computers era view:** Codebreakers or cryptanalysts are **linguistic alchemists** - a mystical tribe attempting to discover meaningful texts in the apparently meaningless sequences of symbols.

WHO ARE CODEBREAKERS - DEVELOPMENTS

The vision of codebreakers has changed through the history, depending on the tools used for encryption and cryptoanalysis.

- **Old times view:** Cryptology is a **black art** and crypanalysts communicate with **dark spirits** and even they are **followers of the devil**.
- **Pre-computers era view:** Codebreakers or cryptanalysts are **linguistic alchemists** - a mystical tribe attempting to discover meaningful texts in the apparently meaningless sequences of symbols.
- **Current view** Codebreakers and cryptanalysts are **artists** that can superbly use modern mathematics, informatics and computing supertechnology for decrypting encrypted messages.

CRYPTO VIEW of MODERN HISTORY

- **First World War** was the war of chemists (deadly gases).

- **First World War** was the war of chemists (deadly gases).
- **Second World War** was the war of physicists (atomic bombs).

- **First World War** was the war of chemists (deadly gases).
- **Second World War** was the war of physicists (atomic bombs).
- **Third World War** will be the war of informaticians (cryptographers and cryptanalysts).

BASIC TYPES of CLASSICAL SECRET-KEY CIPHERS

BASIC TYPES of CLASSICAL SECRET-KEY CIPHERS

Substitution ciphers: are ciphers where units of plaintext are replaced by parts of cryptotext according a fixed rule.

BASIC TYPES of CLASSICAL SECRET-KEY CIPHERS

Substitution ciphers: are ciphers where units of plaintext are replaced by parts of cryptotext according a fixed rule.

Simple substitution ciphers operates on single letters.

BASIC TYPES of CLASSICAL SECRET-KEY CIPHERS

Substitution ciphers: are ciphers where units of plaintext are replaced by parts of cryptotext according a fixed rule.

Simple substitution ciphers operates on single letters.

Monoalphabethic (simple) substitution ciphers: are defined by a single fixed permutation π with encoding

$$e_\pi(a_1 a_2 \dots a_n) = \pi(a_1)\pi(a_2)\dots\pi(a_n)$$

BASIC TYPES of CLASSICAL SECRET-KEY CIPHERS

Substitution ciphers: are ciphers where units of plaintext are replaced by parts of cryptotext according a fixed rule.

Simple substitution ciphers operates on single letters.

Monoalphabethic (simple) substitution ciphers: are defined by a single fixed permutation π with encoding

$$e_\pi(a_1 a_2 \dots a_n) = \pi(a_1)\pi(a_2)\dots\pi(a_n)$$

Polyalphabetic (simple) substitutions systems may use different permutations at different positions of the plaintext.

BASIC TYPES of CLASSICAL SECRET-KEY CIPHERS

Substitution ciphers: are ciphers where units of plaintext are replaced by parts of cryptotext according a fixed rule.

Simple substitution ciphers operates on single letters.

Monoalphabethic (simple) substitution ciphers: are defined by a single fixed permutation π with encoding

$$e_\pi(a_1 a_2 \dots a_n) = \pi(a_1)\pi(a_2)\dots\pi(a_n)$$

Polyalphabetic (simple) substitutions systems may use different permutations at different positions of the plaintext.

Polygraphic (digraphic) substitution ciphers operate on larger, for instance on the length two) substrings of the plaintext.

BASIC TYPES of CLASSICAL SECRET-KEY CIPHERS

Substitution ciphers: are ciphers where units of plaintext are replaced by parts of cryptotext according a fixed rule.

Simple substitution ciphers operates on single letters.

Monoalphabethic (simple) substitution ciphers: are defined by a single fixed permutation π with encoding

$$e_\pi(a_1 a_2 \dots a_n) = \pi(a_1)\pi(a_2)\dots\pi(a_n)$$

Polyalphabetic (simple) substitutions systems may use different permutations at different positions of the plaintext.

Polygraphic (digraphic) substitution ciphers operate on larger, for instance on the length two) substrings of the plaintext.

Transposition ciphers do not replace but only rearrange order of symbols in the plaintext - sometimes in a complicated way.

PARTICULAR CRYPTOSYSTEMS

CAESAR (100 - 42 B.C.) CRYPTOSYSTEM - SHIFT CIPHER I

SHIFT CIPHER is a simple monoalphabetic cipher that can be used to encrypt words in any alphabet.

SHIFT CIPHER is a simple monoalphabetic cipher that can be used to encrypt words in any alphabet.

In order to encrypt words in English alphabet we use:

SHIFT CIPHER is a simple monoalphabetic cipher that can be used to encrypt words in any alphabet.

In order to encrypt words in English alphabet we use:

Key-space: $K = \{1, 2, \ldots, 25\}$

SHIFT CIPHER is a simple monoalphabetic cipher that can be used to encrypt words in any alphabet.

In order to encrypt words in English alphabet we use:

Key-space: $K = \{1, 2, \ldots, 25\}$

For any key $k \in K$, the encryption algorithm e_k for SHIFT CIPHER $SC(k)$ substitutes any letter by the letter occurring k positions ahead (cyclically) in the alphabet.

SHIFT CIPHER is a simple monoalphabetic cipher that can be used to encrypt words in any alphabet.

In order to encrypt words in English alphabet we use:

Key-space: $K = \{1, 2, \ldots, 25\}$

For any key $k \in K$, the encryption algorithm e_k for SHIFT CIPHER $SC(k)$ substitutes any letter by the letter occurring k positions ahead (cyclically) in the alphabet.

The decryption algorithm d_k for $SC(k)$ substitutes any letter by the one occurring k positions backward (cyclically) in the alphabet.

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example $e_2(\text{EXAMPLE}) =$

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example $e_2(\text{EXAMPLE}) = \text{GZCORNG}$,

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example $e_2(\text{EXAMPLE}) = \text{GZCORNG},$
$e_3(\text{EXAMPLE}) =$

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example $e_2(\text{EXAMPLE}) = \text{GZCORNG},$
$e_3(\text{EXAMPLE}) = \text{HADPSOH},$

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example

$$e_2(\text{EXAMPLE}) = \text{GZCORNG},$$
$$e_3(\text{EXAMPLE}) = \text{HADPSOH},$$
$$e_1(\text{HAL}) =$$

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example

$e_2(\text{EXAMPLE}) = \text{GZCORNG}$,
$e_3(\text{EXAMPLE}) = \text{HADPSOH}$,
$e_1(\text{HAL}) = \text{IBM}$,

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example

$e_2(\text{EXAMPLE}) = \text{GZCORNG},$
$e_3(\text{EXAMPLE}) = \text{HADPSOH},$
$e_1(\text{HAL}) = \text{IBM},$
$e_3(\text{COLD}) =$

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example

$e_2(\text{EXAMPLE}) = \text{GZCORNG},$
$e_3(\text{EXAMPLE}) = \text{HADPSOH},$
$e_1(\text{HAL}) = \text{IBM},$
$e_3(\text{COLD}) = \text{FROG}$

ABCDEFGHIJKLMNOPQRSTUVWXYZ

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example $e_2(\text{EXAMPLE}) = \text{GZCORNG}$,
$e_3(\text{EXAMPLE}) = \text{HADPSOH}$,
$e_1(\text{HAL}) = \text{IBM}$,
$e_3(\text{COLD}) = \text{FROG}$

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Example Find the plaintext to the following cryptotext obtained by the encryption with SHIFT CIPHER with $\mathbf{k} = ?$.

Decrypt the cryptotext: **VHFUHW GH GHXA, VHFUHW GH GLHX, VHFUHW GH WURLV, VHFUHW GH WRXV.**

SHIFT CIPHER $SC(k)$ - $SC(3)$ is called CAESAR SHIFT

Example

e_2(EXAMPLE) = GZCORNG,
e_3(EXAMPLE) = HADPSOH,
e_1(HAL) = IBM,
e_3(COLD) = FROG

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Example Find the plaintext to the following cryptotext obtained by the encryption with SHIFT CIPHER with $\mathbf{k} = ?$.

Decrypt the cryptotext: **VHFUHW GH GHXA, VHFUHW GH GLHX, VHFUHW GH WURLV, VHFUHW GH WRXV.**

Numerical version of $SC(k)$ is defined, for English, on the set $\{0, 1, 2, \ldots, 25\}$ by the encryption algorithm:

$$e_k(i) = (i + k)(mod\ 26)$$

Numerical version of the cipher Atbash used in the Bible.

$$e(i) = 25 - i$$

EXAMPLE

EXAMPLE

Decrypt:

EXAMPLE

Decrypt:

VHFUHW GH GHXA
VHFUHW GH GLHX,
VHFUHW GH WURLV
VHFUHW GH WRXV.

EXAMPLE

Decrypt:

VHFUHW GH GHXA
VHFUHW GH GLHX,
VHFUHW GH WURLV
VHFUHW GH WRXV.

Solution:

EXAMPLE

Decrypt:

VHFUHW GH GHXA
VHFUHW GH GLHX,
VHFUHW GH WURLV
VHFUHW GH WRXV.

Solution:

Secret de deux
secret de Dieu,
secret de trois
secret de tous.

VATSYAYANA CIPHER - $SC(2)$

VATSYAYANA CIPHER - $SC(2)$

Vatsyayana was a Hindu philosopher, believed to be the author of Kamasutra and to live in the period 400 BC - 200 BC.

VATSYAYANA CIPHER - $SC(2)$

Vatsyayana was a Hindu philosopher, believed to be the author of Kamasutra and to live in the period 400 BC - 200 BC.

According to his Kamasutra, a girl needs to learn certain arts and certain tricks:

VATSYAYANA CIPHER - $SC(2)$

Vatsyayana was a Hindu philosopher, believed to be the author of Kamasutra and to live in the period 400 BC - 200 BC.

According to his Kamasutra, a girl needs to learn certain arts and certain tricks: to cook,

VATSYAYANA CIPHER - $SC(2)$

Vatsyayana was a Hindu philosopher, believed to be the author of Kamasutra and to live in the period 400 BC - 200 BC.

According to his Kamasutra, a girl needs to learn certain arts and certain tricks: to cook,to read and to write,

VATSYAYANA CIPHER - $SC(2)$

Vatsyayana was a Hindu philosopher, believed to be the author of Kamasutra and to live in the period 400 BC - 200 BC.

According to his Kamasutra, a girl needs to learn certain arts and certain tricks: to cook,to read and to write, and to send her lover secret messages which no one else would be able to decipher.

VATSYAYANA CIPHER - $SC(2)$

Vatsyayana was a Hindu philosopher, believed to be the author of Kamasutra and to live in the period 400 BC - 200 BC.

According to his Kamasutra, a girl needs to learn certain arts and certain tricks: to cook,to read and to write, and to send her lover secret messages which no one else would be able to decipher.

Vatsyayana even described such a cipher which is actually $SC(2)$.

VATSYAYANA CIPHER - $SC(2)$

Vatsyayana was a Hindu philosopher, believed to be the author of Kamasutra and to live in the period 400 BC - 200 BC.

According to his Kamasutra, a girl needs to learn certain arts and certain tricks: to cook,to read and to write, and to send her lover secret messages which no one else would be able to decipher.

Vatsyayana even described such a cipher which is actually $SC(2)$.

This system is now believed, by some, to be the oldest cipher used.

POLYBIOUS CRYPTOSYSTEM - I

POLYBIOUS CRYPTOSYSTEM - I

It is a digraphic cipher developed by Polybious in 2nd century BC.

POLYBIOUS CRYPTOSYSTEM - I

It is a digraphic cipher developed by Polybious in 2nd century BC.
Polybious was a Greek soldier, historian and for 17 years a slave in Rome.

POLYBIOUS CRYPTOSYSTEM - I

It is a digraphic cipher developed by Polybious in 2nd century BC.

Polybious was a Greek soldier, historian and for 17 years a slave in Rome.

POLYBIOUS CRYPTOSYSTEM - II

POLYBIOUS can be used to encrypt words of the English alphabet without J.

POLYBIOUS CRYPTOSYSTEM - II

POLYBIOUS can be used to encrypt words of the English alphabet without J.

Key-space: Polybious checkerboards 5×5 with 25 English letters and with rows + columns labeled by symbols.

POLYBIOUS CRYPTOSYSTEM - II

POLYBIOUS can be used to encrypt words of the English alphabet without J.

Key-space: Polybious checkerboards 5×5 with 25 English letters and with rows + columns labeled by symbols.

Example:

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

Encryption algorithm: Each symbol is substituted by the pair of symbols denoting the row and the column of the checkerboard in which the symbol is placed.

Example: **KONIEC** $\rightarrow$

POLYBIOUS CRYPTOSYSTEM - II

POLYBIOUS can be used to encrypt words of the English alphabet without J.

Key-space: Polybious checkerboards 5×5 with 25 English letters and with rows + columns labeled by symbols.

Example:

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

Encryption algorithm: Each symbol is substituted by the pair of symbols denoting the row and the column of the checkerboard in which the symbol is placed.

Example: **KONIEC** $\rightarrow$BJCI

POLYBIOUS CRYPTOSYSTEM - II

POLYBIOUS can be used to encrypt words of the English alphabet without J.

Key-space: Polybious checkerboards 5×5 with 25 English letters and with rows + columns labeled by symbols.

Example:

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

Encryption algorithm: Each symbol is substituted by the pair of symbols denoting the row and the column of the checkerboard in which the symbol is placed.

Example: **KONIEC** $\rightarrow$BJCICHBI

POLYBIOUS CRYPTOSYSTEM - II

POLYBIOUS can be used to encrypt words of the English alphabet without J.

Key-space: Polybious checkerboards 5×5 with 25 English letters and with rows + columns labeled by symbols.

Example:

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

Encryption algorithm: Each symbol is substituted by the pair of symbols denoting the row and the column of the checkerboard in which the symbol is placed.

Example: **KONIEC** $\rightarrow$BJCICHBIAJAH
Decryption algorithm: ???

FIRST INTERNET

FIRST INTERNET

Observation: Romans were able to created powerful optical information communication networks that allowed them to deliver information and orders very fast along long distances and this way to control efficiently huge territory and to make their armies flexible because they could deliver information and messages much faster than using horses.

FIRST INTERNET

Observation: Romans were able to created powerful optical information communication networks that allowed them to deliver information and orders very fast along long distances and this way to control efficiently huge territory and to make their armies flexible because they could deliver information and messages much faster than using horses.

It is expected that Romans already used Polybious cryptosystem.

KERCHKHOFF's PRINCIPLE

The basic philosophy of modern cryptanalysis is embodied in the following principle formulated in 1883 by **Jean Guillaume Hubert Victor Francois Alexandre Auguste Kerckhoffs von Nieuwenhof** (1835 - 1903).

The basic philosophy of modern cryptanalysis is embodied in the following principle formulated in 1883 by **Jean Guillaume Hubert Victor Francois Alexandre Auguste Kerckhoffs von Nieuwenhof** (1835 - 1903).

The security of a cryptosystem must not depend on keeping secret the encryption algorithm.

KERCKHOFF's PRINCIPLE

The basic philosophy of modern cryptanalysis is embodied in the following principle formulated in 1883 by **Jean Guillaume Hubert Victor Francois Alexandre Auguste Kerckhoffs von Nieuwenhof** (1835 - 1903).

The security of a cryptosystem must not depend on keeping secret the encryption algorithm. The security should depend only on *keeping secret the key.*

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

1. **Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.**

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

1. **Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.**
2. Given d_k and a cryptotext c, it should be **easy** to compute $w = d_k(c)$.

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

1. **Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.**
2. Given d_k and a cryptotext c, it should be **easy** to compute $w = d_k(c)$.
3. **A cryptotext $e_k(w)$ should not be much longer than the plaintext w.**

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

1. **Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.**
2. Given d_k and a cryptotext c, it should be **easy** to compute $w = d_k(c)$.
3. **A cryptotext $e_k(w)$ should not be much longer than the plaintext w.**
4. It should be unfeasible to determine w from $e_k(w)$ without knowing d_k.

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

1. **Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.**
2. Given d_k and a cryptotext c, it should be **easy** to compute $w = d_k(c)$.
3. **A cryptotext $e_k(w)$ should not be much longer than the plaintext w.**
4. It should be unfeasible to determine w from $e_k(w)$ without knowing d_k.
5. **The so called avalanche effect should hold:**

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

1. **Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.**
2. Given d_k and a cryptotext c, it should be **easy** to compute $w = d_k(c)$.
3. **A cryptotext $e_k(w)$ should not be much longer than the plaintext w.**
4. It should be unfeasible to determine w from $e_k(w)$ without knowing d_k.
5. **The so called avalanche effect should hold: A small change in the plaintext, or in the key, should lead to a big change in the cryptotext**

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

1. **Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.**
2. Given d_k and a cryptotext c, it should be easy to compute $w = d_k(c)$.
3. **A cryptotext $e_k(w)$ should not be much longer than the plaintext w.**
4. It should be unfeasible to determine w from $e_k(w)$ without knowing d_k.
5. **The so called avalanche effect should hold: A small change in the plaintext, or in the key, should lead to a big change in the cryptotext (i.e. a change of one bit of the plaintext should result in a change of all bits of the cryptotext, each with the probability close to 0.5).**

BASIC REQUIREMENTS for GOOD CRYPTOSYSTEMS

(Sir Francis R. Bacon (1561 - 1626))

1. **Given e_k and a plaintext w, it should be easy to compute $c = e_k(w)$.**
2. Given d_k and a cryptotext c, it should be **easy** to compute $w = d_k(c)$.
3. **A cryptotext $e_k(w)$ should not be much longer than the plaintext w.**
4. It should be unfeasible to determine w from $e_k(w)$ without knowing d_k.
5. **The so called avalanche effect should hold: A small change in the plaintext, or in the key, should lead to a big change in the cryptotext (i.e. a change of one bit of the plaintext should result in a change of all bits of the cryptotext, each with the probability close to 0.5).**
6. The cryptosystem should **not** be closed under composition, i.e. not for every two keys k_1, k_2 there is a key k such that

$$e_k(w) = e_{k_1}(e_{k_2}(w)).$$

7. **The set of keys should be very large.**

KERKHOFFS' REQUIREMENTS - 1883

KERKHOFFS' REQUIREMENTS - 1883

- Cryptotext should be unbreakable in practice.

KERKHOFFS' REQUIREMENTS - 1883

- Cryptotext should be unbreakable in practice.
- Cryptosystem should be convenient for the correspondence.

KERKHOFFS' REQUIREMENTS - 1883

- Cryptotext should be unbreakable in practice.
- Cryptosystem should be convenient for the correspondence.
- The key should be easily remembered and changeable.

KERKHOFFS' REQUIREMENTS - 1883

- Cryptotext should be unbreakable in practice.
- Cryptosystem should be convenient for the correspondence.
- The key should be easily remembered and changeable.
- The cryptotext should be transmissible by telegraph.

KERKHOFFS' REQUIREMENTS - 1883

- Cryptotext should be unbreakable in practice.
- Cryptosystem should be convenient for the correspondence.
- The key should be easily remembered and changeable.
- The cryptotext should be transmissible by telegraph.
- The cryptosystem apparatus should be easily portable.

KERKHOFFS' REQUIREMENTS - 1883

- Cryptotext should be unbreakable in practice.
- Cryptosystem should be convenient for the correspondence.
- The key should be easily remembered and changeable.
- The cryptotext should be transmissible by telegraph.
- The cryptosystem apparatus should be easily portable.
- The encryption machine should be relatively easy to use.

FOUR DEVELOPMENTS THAT CHANGED METHODS and IMPORTANCE of CRYPTOGRAPHY

- Wide use of telegraph - 1844.

FOUR DEVELOPMENTS THAT CHANGED METHODS and IMPORTANCE of CRYPTOGRAPHY

- Wide use of telegraph - 1844.
- Wide use of radio transmission - 1895.

FOUR DEVELOPMENTS THAT CHANGED METHODS and IMPORTANCE of CRYPTOGRAPHY

- Wide use of telegraph - 1844.
- Wide use of radio transmission - 1895.
- Wide use of encryption/decryption machines - 1930.

FOUR DEVELOPMENTS THAT CHANGED METHODS and IMPORTANCE of CRYPTOGRAPHY

- Wide use of telegraph - 1844.
- Wide use of radio transmission - 1895.
- Wide use of encryption/decryption machines - 1930.
- Wide use of internet.

CRYPTANALYSIS ATTACKS I

The aim of cryptanalysis is to get as much information about the plaintext or the key as possible.

Main types of cryptanalytic attacks

1. **Cryptotexts-only attack.** The cryptanalysts get cryptotexts $c_1 = e_k(w_1), \ldots, c_n = e_k(w_n)$ and try to infer the key k,

CRYPTANALYSIS ATTACKS I

The aim of cryptanalysis is to get as much information about the plaintext or the key as possible.

Main types of cryptanalytic attacks

1. **Cryptotexts-only attack.** The cryptanalysts get cryptotexts $c_1 = e_k(w_1), \dots, c_n = e_k(w_n)$ and try to infer the key k,or as many of the plaintexts $w_1, \dots, w_n$ as possible.
2. **Known-plaintexts attack** (given are some pairs [plaintext, cryptotext]) The cryptanalysts know some pairs $w_i, e_k(w_i), 1 \leq i \leq n$, and try to infer k,

CRYPTANALYSIS ATTACKS I

The aim of cryptanalysis is to get as much information about the plaintext or the key as possible.

Main types of cryptanalytic attacks

1. **Cryptotexts-only attack.** The cryptanalysts get cryptotexts $c_1 = e_k(w_1), \dots, c_n = e_k(w_n)$ and try to infer the key k,or as many of the plaintexts $w_1, \dots, w_n$ as possible.
2. **Known-plaintexts attack** (given are some pairs [plaintext, cryptotext]) The cryptanalysts know some pairs $w_i, e_k(w_i), 1 \leq i \leq n$, and try to infer k, or at least w_{n+1} for a new cryptotext $e_k(w_{n+1})$.
3. **Chosen-plaintexts attack** (given are cryptotext for some chosen plaintexts).

CRYPTANALYSIS ATTACKS I

The aim of cryptanalysis is to get as much information about the plaintext or the key as possible.

Main types of cryptanalytic attacks

1. **Cryptotexts-only attack.** The cryptanalysts get cryptotexts $c_1 = e_k(w_1), \ldots, c_n = e_k(w_n)$ and try to infer the key k,or as many of the plaintexts $w_1, \ldots, w_n$ as possible.
2. **Known-plaintexts attack** (given are some pairs [plaintext, cryptotext]) The cryptanalysts know some pairs $w_i, e_k(w_i), 1 \leq i \leq n$, and try to infer k, or at least w_{n+1} for a new cryptotext $e_k(w_{n+1})$.
3. **Chosen-plaintexts attack** (given are cryptotext for some chosen plaintexts). The cryptanalysts choose plaintexts $w_1, \ldots, w_n$ to get cryptotexts $e_k(w_1), \ldots, e_k(w_n)$, and try to infer k

CRYPTANALYSIS ATTACKS I

The aim of cryptanalysis is to get as much information about the plaintext or the key as possible.

Main types of cryptanalytic attacks

1. **Cryptotexts-only attack.** The cryptanalysts get cryptotexts $c_1 = e_k(w_1), \ldots, c_n = e_k(w_n)$ and try to infer the key k,or as many of the plaintexts $w_1, \ldots, w_n$ as possible.
2. **Known-plaintexts attack** (given are some pairs [plaintext, cryptotext]) The cryptanalysts know some pairs $w_i, e_k(w_i), 1 \leq i \leq n$, and try to infer k, or at least w_{n+1} for a new cryptotext $e_k(w_{n+1})$.
3. **Chosen-plaintexts attack** (given are cryptotext for some chosen plaintexts). The cryptanalysts choose plaintexts $w_1, \ldots, w_n$ to get cryptotexts $e_k(w_1), \ldots, e_k(w_n)$, and try to infer k or at least w_{n+1} for a new cryptotext $c_{n+1} = e_k(w_{n+1})$.

CRYPTANALYSIS ATTACKS I

The aim of cryptanalysis is to get as much information about the plaintext or the key as possible.

Main types of cryptanalytic attacks

1. **Cryptotexts-only attack.** The cryptanalysts get cryptotexts $c_1 = e_k(w_1), \ldots, c_n = e_k(w_n)$ and try to infer the key k,or as many of the plaintexts $w_1, \ldots, w_n$ as possible.
2. **Known-plaintexts attack** (given are some pairs [plaintext, cryptotext]) The cryptanalysts know some pairs $w_i, e_k(w_i), 1 \leq i \leq n$, and try to infer k, or at least w_{n+1} for a new cryptotext $e_k(w_{n+1})$.
3. **Chosen-plaintexts attack** (given are cryptotext for some chosen plaintexts). The cryptanalysts choose plaintexts $w_1, \ldots, w_n$ to get cryptotexts $e_k(w_1), \ldots, e_k(w_n)$, and try to infer k or at least w_{n+1} for a new cryptotext $c_{n+1} = e_k(w_{n+1})$. (For example, if they get temporary access to the encryption machinery.)

CRYPTANALYSIS ATTACKS - II.

4. **Known-encryption-algorithm attack**
   The encryption algorithm e_k is given and the cryptanalysts try to get the decryption algorithm d_k.

CRYPTANALYSIS ATTACKS - II.

4. **Known-encryption-algorithm attack**
   The encryption algorithm e_k is given and the cryptanalysts try to get the decryption algorithm d_k.

5. **Chosen-cryptotext attack (given are plaintexts for some chosen cryptotexts)**
   The cryptanalysts know some pairs

$$[c_i, d_k(c_i)], \quad 1 \leq i \leq n,$$

   where the cryptotexts c_i have been chosen by the cryptanalysts. The aim is to determine the key. (For example, if cryptanalysts get a temporary access to decryption machinery.)

WHAT CAN BAD EVE DO?

WHAT CAN BAD EVE DO?

Let us assume that a clever Alice sends an encrypted message to Bob.

WHAT CAN BAD EVE DO?

Let us assume that a clever Alice sends an encrypted message to Bob.
What can a bad enemy, called usually Eve (eavesdropper), do?

WHAT CAN BAD EVE DO?

Let us assume that a clever Alice sends an encrypted message to Bob.
What can a bad enemy, called usually Eve (eavesdropper), do?

- Eve can read (and try to decrypt) the message.

WHAT CAN BAD EVE DO?

Let us assume that a clever Alice sends an encrypted message to Bob.
What can a bad enemy, called usually Eve (eavesdropper), do?

- Eve can read (and try to decrypt) the message.
- Eve can try to get the key that was used and then decrypt all messages encrypted with the same key.

WHAT CAN BAD EVE DO?

Let us assume that a clever Alice sends an encrypted message to Bob.
What can a bad enemy, called usually Eve (eavesdropper), do?

- Eve can read (and try to decrypt) the message.
- Eve can try to get the key that was used and then decrypt all messages encrypted with the same key.
- Eve can change the message sent by Alice into another message, in such a way that Bob will have the feeling, after he gets the changed message, that it was a message from Alice.

WHAT CAN BAD EVE DO?

Let us assume that a clever Alice sends an encrypted message to Bob.
What can a bad enemy, called usually Eve (eavesdropper), do?

- Eve can read (and try to decrypt) the message.
- Eve can try to get the key that was used and then decrypt all messages encrypted with the same key.
- Eve can change the message sent by Alice into another message, in such a way that Bob will have the feeling, after he gets the changed message, that it was a message from Alice.
- Eve can pretend to be Alice and communicate with Bob, in such a way that Bob thinks he is communicating with Alice.

WHAT CAN BAD EVE DO?

Let us assume that a clever Alice sends an encrypted message to Bob.
What can a bad enemy, called usually Eve (eavesdropper), do?

- Eve can read (and try to decrypt) the message.
- Eve can try to get the key that was used and then decrypt all messages encrypted with the same key.
- Eve can change the message sent by Alice into another message, in such a way that Bob will have the feeling, after he gets the changed message, that it was a message from Alice.
- Eve can pretend to be Alice and communicate with Bob, in such a way that Bob thinks he is communicating with Alice.

An eavesdropper can therefore be passive - Eve or active - Mallot.

BASIC GOALS of BROADLY UNDERSTOOD CRYPTOGRAPHY

BASIC GOALS of BROADLY UNDERSTOOD CRYPTOGRAPHY

Confidentiality: Eve should not be able to decrypt the message Alice sends to Bob.

BASIC GOALS of BROADLY UNDERSTOOD CRYPTOGRAPHY

Confidentiality: Eve should not be able to decrypt the message Alice sends to Bob.

Data integrity: Bob wants to be sure that Alice's message has not been altered by Eve.

BASIC GOALS of BROADLY UNDERSTOOD CRYPTOGRAPHY

Confidentiality: Eve should not be able to decrypt the message Alice sends to Bob.

Data integrity: Bob wants to be sure that Alice's message has not been altered by Eve.

Authentication: Bob wants to be sure that only Alice could have sent the message he has received.

BASIC GOALS of BROADLY UNDERSTOOD CRYPTOGRAPHY

Confidentiality: Eve should not be able to decrypt the message Alice sends to Bob.

Data integrity: Bob wants to be sure that Alice's message has not been altered by Eve.

Authentication: Bob wants to be sure that only Alice could have sent the message he has received.

Non-repudiation: Alice should not be able to claim that she did not send messages that she has sent.

BASIC GOALS of BROADLY UNDERSTOOD CRYPTOGRAPHY

Confidentiality: Eve should not be able to decrypt the message Alice sends to Bob.

Data integrity: Bob wants to be sure that Alice's message has not been altered by Eve.

Authentication: Bob wants to be sure that only Alice could have sent the message he has received.

Non-repudiation: Alice should not be able to claim that she did not send messages that she has sent.

Anonymity: Alice does not want Bob to find out who sent the message

HILL CRYPTOSYSTEM I

HILL CRYPTOSYSTEM I

The polygraphic cryptosystem presented in this slide was probably never used.

HILL CRYPTOSYSTEM I

The polygraphic cryptosystem presented in this slide was probably never used. In spite of that this cryptosystem played an important role in the history of modern cryptography.

HILL CRYPTOSYSTEM I

The polygraphic cryptosystem presented in this slide was probably never used. In spite of that this cryptosystem played an important role in the history of modern cryptography.

We describe Hill cryptosystem for a fixed n and the English alphabet.

HILL CRYPTOSYSTEM I

The polygraphic cryptosystem presented in this slide was probably never used. In spite of that this cryptosystem played an important role in the history of modern cryptography.

We describe Hill cryptosystem for a fixed n and the English alphabet.

Key-space: The set of all matrices M of degree n with elements from the set $\{0, 1, \ldots, 25\}$ such that $M^{-1} mod\ 26$ exists.

HILL CRYPTOSYSTEM I

The polygraphic cryptosystem presented in this slide was probably never used. In spite of that this cryptosystem played an important role in the history of modern cryptography.

We describe Hill cryptosystem for a fixed n and the English alphabet.

Key-space: The set of all matrices M of degree n with elements from the set $\{0, 1, \ldots, 25\}$ such that $M^{-1} mod\ 26$ exists.

Plaintext + cryptotext space: English words of length n.

HILL CRYPTOSYSTEM I

The polygraphic cryptosystem presented in this slide was probably never used. In spite of that this cryptosystem played an important role in the history of modern cryptography.

We describe Hill cryptosystem for a fixed n and the English alphabet.

Key-space: The set of all matrices M of degree n with elements from the set $\{0, 1, \ldots, 25\}$ such that $M^{-1} mod\ 26$ exists.

Plaintext + cryptotext space: English words of length n.

Encoding: For a word w let c_w be the column vector of length n of the integer codes of symbols of w. $(A \to 0, B \to 1, C \to 2, \ldots)$

Encryption: $c_c = Mc_w$ mod 26

Decryption: $c_w = M^{-1}c_c$ mod 26

HILL CRYPTOSYSTEM - EXAMPLE

HILL CRYPTOSYSTEM - EXAMPLE

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

HILL CRYPTOSYSTEM - EXAMPLE

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

$$M = \begin{bmatrix} 4 & 7 \\ 1 & 1 \end{bmatrix} \quad M^{-1} = \begin{bmatrix} 17 & 11 \\ 9 & 16 \end{bmatrix}$$

HILL CRYPTOSYSTEM - EXAMPLE

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

$$M = \begin{bmatrix} 4 & 7 \\ 1 & 1 \end{bmatrix} \quad M^{-1} = \begin{bmatrix} 17 & 11 \\ 9 & 16 \end{bmatrix}$$

Plaintext: $w = \text{LONDON}$

HILL CRYPTOSYSTEM - EXAMPLE

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

$$M = \begin{bmatrix} 4 & 7 \\ 1 & 1 \end{bmatrix} \quad M^{-1} = \begin{bmatrix} 17 & 11 \\ 9 & 16 \end{bmatrix}$$

Plaintext: $w =$ LONDON

Encodings: $w_{LO} = \begin{bmatrix} 11 \\ 14 \end{bmatrix}, \quad w_{ND} = \begin{bmatrix} 13 \\ 3 \end{bmatrix}, w_{ON} = \begin{bmatrix} 14 \\ 13 \end{bmatrix}$

Encryption : $Mw_{LO} = \begin{bmatrix} 12 \\ 25 \end{bmatrix}, \quad Mw_{ND} = \begin{bmatrix} 21 \\ 16 \end{bmatrix}, Mw_{ON} = \begin{bmatrix} 17 \\ 1 \end{bmatrix}$

HILL CRYPTOSYSTEM - EXAMPLE

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

$$M = \begin{bmatrix} 4 & 7 \\ 1 & 1 \end{bmatrix} \quad M^{-1} = \begin{bmatrix} 17 & 11 \\ 9 & 16 \end{bmatrix}$$

Plaintext: w = LONDON

$$\textbf{Encodings:} \quad w_{LO} = \begin{bmatrix} 11 \\ 14 \end{bmatrix}, \quad w_{ND} = \begin{bmatrix} 13 \\ 3 \end{bmatrix}, w_{ON} = \begin{bmatrix} 14 \\ 13 \end{bmatrix}$$

$$\textbf{Encryption}: \quad Mw_{LO} = \begin{bmatrix} 12 \\ 25 \end{bmatrix}, \quad Mw_{ND} = \begin{bmatrix} 21 \\ 16 \end{bmatrix}, Mw_{ON} = \begin{bmatrix} 17 \\ 1 \end{bmatrix}$$

Cryptotext: MZVQRB

HILL CRYPTOSYSTEM - EXAMPLE

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

$$M = \begin{bmatrix} 4 & 7 \\ 1 & 1 \end{bmatrix} \quad M^{-1} = \begin{bmatrix} 17 & 11 \\ 9 & 16 \end{bmatrix}$$

Plaintext: $w =$ LONDON

Encodings: $w_{LO} = \begin{bmatrix} 11 \\ 14 \end{bmatrix}, \quad w_{ND} = \begin{bmatrix} 13 \\ 3 \end{bmatrix}, w_{ON} = \begin{bmatrix} 14 \\ 13 \end{bmatrix}$

Encryption : $Mw_{LO} = \begin{bmatrix} 12 \\ 25 \end{bmatrix}, \quad Mw_{ND} = \begin{bmatrix} 21 \\ 16 \end{bmatrix}, Mw_{ON} = \begin{bmatrix} 17 \\ 1 \end{bmatrix}$

Cryptotext: MZVQRB

Theorem

$$\text{If } M = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \text{ then } M^{-1} = \frac{1}{\det M} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

Proof: Exercise

INVERTING INTEGER MATRICES modulo n

INVERTING INTEGER MATRICES modulo n

The basic idea to compute M^{-1} (mod n) is simple:

INVERTING INTEGER MATRICES modulo n

The basic idea to compute $M^{-1} \pmod{n}$ is simple:

Use the usual method to invert M in terms of rational numbers, and then replace each a/b by ab^{-1}, where $bb^{-1} \equiv 1 \pmod{n}$.

INVERTING INTEGER MATRICES modulo n

The basic idea to compute $M^{-1} \pmod{n}$ is simple:

Use the usual method to invert M in terms of rational numbers, and then replace each a/b by ab^{-1}, where $bb^{-1} \equiv 1 \pmod{n}$.

Example: Compute the inverse of the following matrix modulo 11:

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 9 \end{pmatrix} \pmod{11}.$$

The standard inverse of M in rational numbers is

$$\frac{1}{2}\begin{pmatrix} 6 & -5 & 1 \\ -6 & 8 & -2 \\ 2 & -3 & 1 \end{pmatrix}$$

INVERTING INTEGER MATRICES modulo n

The basic idea to compute $M^{-1} \pmod{n}$ is simple:

Use the usual method to invert M in terms of rational numbers, and then replace each a/b by ab^{-1}, where $bb^{-1} \equiv 1 \pmod{n}$.

Example: Compute the inverse of the following matrix modulo 11:

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 4 & 9 \end{pmatrix} \pmod{11}.$$

The standard inverse of M in rational numbers is

$$\frac{1}{2}\begin{pmatrix} 6 & -5 & 1 \\ -6 & 8 & -2 \\ 2 & -3 & 1 \end{pmatrix}$$

Since $2^{-1} \equiv 6 \pmod{11}$, the resulting matrix has the form

$$M^{-1} = \begin{pmatrix} 3 & 3 & 6 \\ 8 & 4 & 10 \\ 1 & 4 & 6 \end{pmatrix} \pmod{11}.$$

SESTER S. HILL

Hill published his cryptosystem, based on the ideas of Giovani Bathista Porta (1535-1615), in the paper

Cryptography in an algebraic alphabet

Hill published his cryptosystem, based on the ideas of Giovani Bathista Porta (1535-1615), in the paper

Cryptography in an algebraic alphabet

in the journal **American Mathematical Monthly** in 1929.

Hill even tried to design a machine to use his cipher, but without a success.

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS

A cryptosystem is called secret-key cryptosystem if some secret piece of information – the key – has to be agreed first between any two parties that have, or want, to communicate through the cryptosystem.

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS

A cryptosystem is called secret-key cryptosystem if some secret piece of information – the key – has to be agreed first between any two parties that have, or want, to communicate through the cryptosystem. Example: **CAESAR, HILL**. Another name is **symmetric cryptosystem (cryptography).**

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS

A cryptosystem is called secret-key cryptosystem if some secret piece of information – the key – has to be agreed first between any two parties that have, or want, to communicate through the cryptosystem. Example: **CAESAR, HILL**.
Another name is **symmetric cryptosystem (cryptography).**

Two basic types of secret-key cryptosystems

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS

A cryptosystem is called secret-key cryptosystem if some secret piece of information – the key – has to be agreed first between any two parties that have, or want, to communicate through the cryptosystem. Example: CAESAR, HILL. Another name is **symmetric cryptosystem (cryptography).**

Two basic types of secret-key cryptosystems

- **substitution** based cryptosystems

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS

A cryptosystem is called secret-key cryptosystem if some secret piece of information – the key – has to be agreed first between any two parties that have, or want, to communicate through the cryptosystem. Example: CAESAR, HILL. Another name is **symmetric cryptosystem (cryptography).**

Two basic types of secret-key cryptosystems

- **substitution** based cryptosystems
- **transposition** based cryptosystems

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS

A cryptosystem is called secret-key cryptosystem if some secret piece of information – the key – has to be agreed first between any two parties that have, or want, to communicate through the cryptosystem. Example: CAESAR, HILL. Another name is **symmetric cryptosystem (cryptography).**

Two basic types of secret-key cryptosystems

- **substitution** based cryptosystems
- **transposition** based cryptosystems

Two basic types of substitution cryptosystems

- **monoalphabetic cryptosystems** – they use a fixed substitution – CAESAR, POLYBIOUS
- **polyalphabetic cryptosystems** – substitution keeps changing during the encryption

SECRET-KEY (SYMMETRIC) CRYPTOSYSTEMS

A cryptosystem is called secret-key cryptosystem if some secret piece of information – the key – has to be agreed first between any two parties that have, or want, to communicate through the cryptosystem. Example: CAESAR, HILL. Another name is **symmetric cryptosystem (cryptography).**

Two basic types of secret-key cryptosystems

- **substitution** based cryptosystems
- **transposition** based cryptosystems

Two basic types of substitution cryptosystems

- **monoalphabetic cryptosystems** – they use a fixed substitution – CAESAR, POLYBIOUS
- **polyalphabetic cryptosystems** – substitution keeps changing during the encryption

A monoalphabetic cryptosystem with letter-by-letter substitution is uniquely specified by a permutation of letters, (number of permutations (keys) is 26!)

AFFINE CRYPTOSYSTEMS

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5,$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, \quad e_{3,5}(x) = (3x + 5) \bmod 26,$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, \quad e_{3,5}(x) = (3x + 5) \bmod 26,$
$e_{3,5}(3) =$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, \quad e_{3,5}(x) = (3x + 5) \bmod 26,$
$e_{3,5}(3) = 14,$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, \quad e_{3,5}(x) = (3x + 5) \bmod 26,$
$e_{3,5}(3) = 14, \quad e_{3,5}(15) =$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, \quad e_{3,5}(x) = (3x + 5) \bmod 26,$
$e_{3,5}(3) = 14, \quad e_{3,5}(15) = 24, \quad e_{3,5}(D) =$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, \quad e_{3,5}(x) = (3x + 5) \bmod 26,$
$e_{3,5}(3) = 14, \quad e_{3,5}(15) = 24, \quad e_{3,5}(D) = O, e_{3,5}(P) =$

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, \quad e_{3,5}(x) = (3x + 5) \bmod 26,$
$e_{3,5}(3) = 14, \quad e_{3,5}(15) = 24, \quad e_{3,5}(D) = O, e_{3,5}(P) = Y$

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

AFFINE CRYPTOSYSTEMS

Example: Each **AFFINE cryptosystem** is given by two integers

$$0 \leq a, b \leq 25, gcd(a, 26) = 1.$$

Encryption: $e_{a,b}(x) = (ax + b) \bmod 26$

Example

$a = 3, b = 5, \quad e_{3,5}(x) = (3x + 5) \bmod 26,$
$e_{3,5}(3) = 14, \quad e_{3,5}(15) = 24, \quad e_{3,5}(D) = O, e_{3,5}(P) = Y$

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Decryption: $d_{a,b}(y) = a^{-1}(y - b) \bmod 26$

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:**

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:** the number of each letter in the cryptotext is counted.

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:** the number of each letter in the cryptotext is counted. **The distributions of letters in the cryptotext is then compared with some official distribution of letters in the plaintext language.**

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:** the number of each letter in the cryptotext is counted. **The distributions of letters in the cryptotext is then compared with some official distribution of letters in the plaintext language.**

The letter with the highest frequency in the cryptotext is likely to be the substitute for the letter with highest frequency in the plaintext language . . . .

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:** the number of each letter in the cryptotext is counted. **The distributions of letters in the cryptotext is then compared with some official distribution of letters in the plaintext language.**

The letter with the highest frequency in the cryptotext is likely to be the substitute for the letter with highest frequency in the plaintext language . . . . The likelihood grows with the length of cryptotext.

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:** the number of each letter in the cryptotext is counted. **The distributions of letters in the cryptotext is then compared with some official distribution of letters in the plaintext language.**

The letter with the highest frequency in the cryptotext is likely to be the substitute for the letter with highest frequency in the plaintext language . . . . The likelihood grows with the length of cryptotext.

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:** the number of each letter in the cryptotext is counted. **The distributions of letters in the cryptotext is then compared with some official distribution of letters in the plaintext language.**

The letter with the highest frequency in the cryptotext is likely to be the substitute for the letter with highest frequency in the plaintext language . . . . The likelihood grows with the length of cryptotext.

Frequency counts in English:

	%		%		%
E	12.31	L	4.03	B	1.62
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		
	70.02		24.71		5.27

and for other languages:

English	%	German	%	Finnish	%	French	%	Italian	%	Spanish	%
E	12.31	E	18.46	A	12.06	E	15.87	E	11.79	E	13.15
T	9.59	N	11.42	I	10.59	A	9.42	A	11.74	A	12.69
A	8.05	I	8.02	T	9.76	I	8.41	I	11.28	O	9.49
O	7.94	R	7.14	N	8.64	S	7.90	O	9.83	S	7.60
N	7.19	S	7.04	E	8.11	T	7.29	N	6.88	N	6.95
I	7.18	A	5.38	S	7.83	N	7.15	L	6.51	R	6.25
S	6.59	T	5.22	L	5.86	R	6.46	R	6.37	I	6.25
R	6.03	U	5.01	O	5.54	U	6.24	T	5.62	L	5.94
H	5.14	D	4.94	K	5.20	L	5.34	S	4.98	D	5.58

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:** the number of each letter in the cryptotext is counted. **The distributions of letters in the cryptotext is then compared with some official distribution of letters in the plaintext language.**

The letter with the highest frequency in the cryptotext is likely to be the substitute for the letter with highest frequency in the plaintext language . . . . The likelihood grows with the length of cryptotext.

Frequency counts in English:

	%		%		%
E	12.31	L	4.03	B	1.62
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		
	70.02		24.71		5.27

and for other languages:

English	%	German	%	Finnish	%	French	%	Italian	%	Spanish	%
E	12.31	E	18.46	A	12.06	E	15.87	E	11.79	E	13.15
T	9.59	N	11.42	I	10.59	A	9.42	A	11.74	A	12.69
A	8.05	I	8.02	T	9.76	I	8.41	I	11.28	O	9.49
O	7.94	R	7.14	N	8.64	S	7.90	O	9.83	S	7.60
N	7.19	S	7.04	E	8.11	T	7.29	N	6.88	N	6.95
I	7.18	A	5.38	S	7.83	N	7.15	L	6.51	R	6.25
S	6.59	T	5.22	L	5.86	R	6.46	R	6.37	I	6.25
R	6.03	U	5.01	O	5.54	U	6.24	T	5.62	L	5.94
H	5.14	D	4.94	K	5.20	L	5.34	S	4.98	D	5.58

The 20 most common digrams are (in decreasing order) TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS.

CRYPTANALYSIS

The basic cryptanalytic attack against monoalphabetic substitution cryptosystems begins with a so called **frequency count:** the number of each letter in the cryptotext is counted. **The distributions of letters in the cryptotext is then compared with some official distribution of letters in the plaintext language.**

The letter with the highest frequency in the cryptotext is likely to be the substitute for the letter with highest frequency in the plaintext language . . . . The likelihood grows with the length of cryptotext.

Frequency counts in English:

	%		%		%
E	12.31	L	4.03	B	1.62
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		
	70.02		24.71		5.27

and for other languages:

English	%	German	%	Finnish	%	French	%	Italian	%	Spanish	%
E	12.31	E	18.46	A	12.06	E	15.87	E	11.79	E	13.15
T	9.59	N	11.42	I	10.59	A	9.42	A	11.74	A	12.69
A	8.05	I	8.02	T	9.76	I	8.41	I	11.28	O	9.49
O	7.94	R	7.14	N	8.64	S	7.90	O	9.83	S	7.60
N	7.19	S	7.04	E	8.11	T	7.29	N	6.88	N	6.95
I	7.18	A	5.38	S	7.83	N	7.15	L	6.51	R	6.25
S	6.59	T	5.22	L	5.86	R	6.46	R	6.37	I	6.25
R	6.03	U	5.01	O	5.54	U	6.24	T	5.62	L	5.94
H	5.14	D	4.94	K	5.20	L	5.34	S	4.98	D	5.58

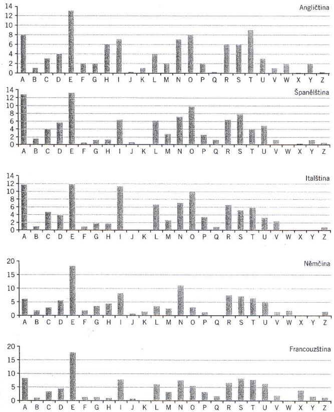
The 20 most common digrams are (in decreasing order) TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS. The six most common **trigrams** are: THE, ING, AND, HER, ERE, ENT.

FREQUENCY ANALYSIS for SEVERAL LANGUAGES

FREQUENCY ANALYSIS for SEVERAL LANGUAGES

NEJČETNĚJŠÍ PÍSMENA V ZÁPADOEVROPSKÝCH JAZYCÍCH

Angličtina:	E T A O I N S H R D L U
Francouzština:	E N A S R I U T O L D C
Němčina:	E N R I S T U D A H G L
Italština:	E I A O R L N T S C D P
Španělština:	E A O S R I N L D C T U



OTHER CHARACTERISTICS of ENGLISH

OTHER CHARACTERISTICS of ENGLISH

V ANGLIČTINĚ

Nejčastější písmena:	**e t a o i n s h r d l u**
Nejčastější první písmena:	**t a s o i c p b s h m**
Nejčastější poslední písmena:	**e t s d n r y o f l a g**
Nejčastější dvojice písmen:	**th er on an re he in ed nd ha at**
Nejčastější trojice písmen:	**the and tha ent ion tio for nde**
Nejčastější zdvojení písmen:	**ss ee tt ff ll mm oo**
Nejčastější písmena následující po E:	**r d s n a c t m e p w o**
Nejčastější dvojpísmenná slova:	**of to in it is be as at so we he**
Nejčastější trojpísmenná slova:	**the and for are but not you all**
Nejčastější čtyřpísmenná slova:	**that with have this will your from they**

FREQUENCY COUNTS in CZECH and SLOVAK

First resource

Czech		*Slovak*	
o	8.66	*a*	10.67
e	7.69	*o*	9.12
n	6.53	*e*	8.43
a	6.21	*i*	5.74
t	5.72	*n*	5.74
v	4.66	*s*	5.02
s	4.51	*t*	4.92
i	4.35	*v*	4.60
l	3.84	*k*	3.96

Second resource:

Czech		*Slovak*	
e	10.13	*a*	9.49
a	8.99	*o*	9.34
o	8.39	*e*	9.16
i	6.92	*i*	6.81
n	6.64	*n*	6.34
s	5.74	*s*	5.94
r	5.33	*r*	5.12
t	4.98	*t*	5.06
v	4.50	*v*	4.85

Discovery of FREQUENCY ANALYSIS - I.

Discovery of FREQUENCY ANALYSIS - I.

It was discovered, in 1987, that this technique was already described in 9th century in

Discovery of FREQUENCY ANALYSIS - I.

It was discovered, in 1987, that this technique was already described in 9th century in

a manuscript on deciphering cryptographic messages

Discovery of FREQUENCY ANALYSIS - I.

It was discovered, in 1987, that this technique was already described in 9th century in

a manuscript on deciphering cryptographic messages

written by the" philosopher of the Arabs",called

Discovery of FREQUENCY ANALYSIS - I.

It was discovered, in 1987, that this technique was already described in 9th century in

a manuscript on deciphering cryptographic messages

written by the" philosopher of the Arabs",called

Abú Yúsúf Ya'qúb ibn Is-háq ibn as-Sabbáh ibn 'omrán ibn Ismail a-Kindi

Discovery of FREQUENCY ANALYSIS - I.

It was discovered, in 1987, that this technique was already described in 9th century in

a manuscript on deciphering cryptographic messages

written by the" philosopher of the Arabs",called

Abú Yúsúf Ya'qúb ibn Is-háq ibn as-Sabbáh ibn 'omrán ibn Ismail a-Kindi

He wrote 290 books on medicine, astronomy, mathematics, music,...

Discovery of FREQUENCY ANALYSIS - I.

It was discovered, in 1987, that this technique was already described in 9th century in

a manuscript on deciphering cryptographic messages

written by the" philosopher of the Arabs",called

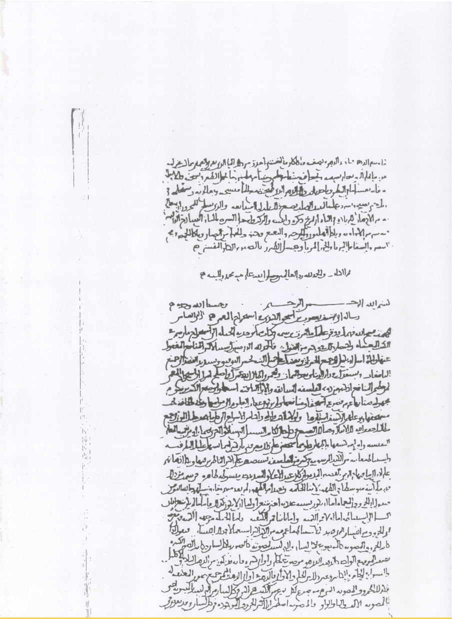
Abú Yúsúf Ya'qúb ibn Is-háq ibn as-Sabbáh ibn 'omrán ibn Ismail a-Kindi

He wrote 290 books on medicine, astronomy, mathematics, music,...

Frequency analysis was originally used to study Koran, to establish chronology of revelations by Muhammad in Koran.

Discovery of FREQUENCY ANALYSIS - II.

Discovery of FREQUENCY ANALYSIS - II.



CRYPTANALYSIS of AFFINE CRYPTOSYSTEM - EXAMPLE

CRYPTANALYSIS of AFFINE CRYPTOSYSTEM - EXAMPLE

Cryptanalysis of a cryptotext encrypted using the AFFINE cryptosystem with an encryption algorithm

$$e_{a,b}(x) = (ax + b) \bmod 26$$

where $0 \leq a, b \leq 25, gcd(a, 26) = 1$. (Number of keys: $12 \times 26 = 312$.)

CRYPTANALYSIS of AFFINE CRYPTOSYSTEM - EXAMPLE

Cryptanalysis of a cryptotext encrypted using the AFFINE cryptosystem with an encryption algorithm

$$e_{a,b}(x) = (ax + b) \bmod 26$$

where $0 \leq a, b \leq 25, gcd(a, 26) = 1$. (Number of keys: $12 \times 26 = 312$.)

Example: Assume that an English plaintext is divided into blocks of 5 letters and encrypted by an AFFINE cryptosystem (ignoring space and interpunctions) as follows:

How to find the plaintext?

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

CRYPTANALYSIS - CONTINUATION I

Frequency analysis of plaintext and frequency table for English:

X - 32	J - 11	D - 2
U - 30	O - 6	V - 2
H - 23	R - 6	F - 1
B - 19	G - 5	P - 1
L - 19	M - 4	E - 0
N - 16	Y - 4	I - 0
K - 15	Z - 4	Q - 0
S - 15	C - 3	T - 0
W - 14	A - 2	

	%		%		%
E	12.31	L	4.03	B	1.62
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		
	70.02		24.71		5.27

First guess: $E = X, T = U$

Encodings: $4a + b = 23 \pmod{26}$

$xa + b = y$ $\quad 19a + b = 20 \pmod{26}$

Solutions: $a = 5, b = 3 \rightarrow a^{-1} =$

CRYPTANALYSIS - CONTINUATION I

Frequency analysis of plaintext and frequency table for English:

X - 32	J - 11	D - 2
U - 30	O - 6	V - 2
H - 23	R - 6	F - 1
B - 19	G - 5	P - 1
L - 19	M - 4	E - 0
N - 16	Y - 4	I - 0
K - 15	Z - 4	Q - 0
S - 15	C - 3	T - 0
W - 14	A - 2	

	%		%		%
E	12.31	L	4.03	B	1.62
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		
	70.02		24.71		5.27

First guess: $E = X, T = U$

Encodings: $xa + b = y$

$$4a + b = 23 \pmod{26}$$

$$19a + b = 20 \pmod{26}$$

Solutions: $a = 5, b = 3 \rightarrow a^{-1} = 21$

CRYPTANALYSIS - CONTINUATION I

Frequency analysis of plaintext and frequency table for English:

X - 32	J - 11	D - 2
U - 30	O - 6	V - 2
H - 23	R - 6	F - 1
B - 19	G - 5	P - 1
L - 19	M - 4	E - 0
N - 16	Y - 4	I - 0
K - 15	Z - 4	Q - 0
S - 15	C - 3	T - 0
W - 14	A - 2	

	%		%		%
E	12.31	L	4.03	B	1.62
T	9.59	D	3.65	G	1.61
A	8.05	C	3.20	V	0.93
O	7.94	U	3.10	K	0.52
N	7.19	P	2.29	Q	0.20
I	7.18	F	2.28	X	0.20
S	6.59	M	2.25	J	0.10
R	6.03	W	2.03	Z	0.09
H	5.14	Y	1.88		
	70.02		24.71		5.27

First guess: $E = X, T = U$

Encodings: $4a + b = 23 \pmod{26}$

$xa + b = y$ $\quad 19a + b = 20 \pmod{26}$

Solutions: $a = 5, b = 3 \rightarrow a^{-1} = 21$

Translation table

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

```
B H J U H   N B U L S   V U L R U   S L Y X H
O N U U N   B W N U A   X U S N L   U Y J S S
W X R L K   G N B O N   U U N B W   S W X K X
H K X D H   U Z D L K   X B H J U   H B N U O
N U M H U   G S W H U   X M B X R   W X K X L
U X B H J   U H C X K   X A X K Z   S W K X X
L K O L J   K C X L C   M X O N U   U B V U L
R R W H S   H B H J U   H N B X M   B X R W X
K X N O Z   L J B X X   H B N F U   B H J U H
L U S W X   G L L K Z   L J P H U   U L S Y X
B J K X S   W H S S W   X K X N B   H B H J U
H Y X W N   U G S W X   G L L K
```

provides from the above cryptotext the plaintext that starts with KGWTG CKTMO OTMIT DMZEG, which does not make sense.

CRYPTANALYSIS - CONTINUATION II

Second guess: $E = X, A = H$

Equations
$$4a + b = 23 \pmod{26}$$
$$b = 7 \pmod{26}$$

Solutions: $a = 4$ or $a = 17$ and therefore $a = 17$

Second guess: $E = X, A = H$

Equations
$$4a + b = 23 \pmod{26}$$
$$b = 7 \pmod{26}$$

Solutions: $a = 4$ or $a = 17$ and therefore $a = 17$

This gives the translation table

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

and the following plaintext from the above cryptotext

SAUNA ISNOT KNOWN TOBEA
FINNI SHINV ENTIO NBUTT
HEWOR DISFI NNISH THERE
AREMA NYMOR ESAUN ASINF
INLAN DTHAN ELSEW HEREO
NESAU NAPER EVERY THREE
ORFOU RPEOP LEFIN NSKNO
WWHAT ASAUN AISEL SEWHE
REIFY OUSEE ASIGN SAUNA
ONTHE DOORY OUCAN NOTBE
SURET HATTH EREIS ASAUN
ABEHI NDTHE DOOR

OTHER EXAMPLES of MONOALPHABETIC CRYPTOSYSTEMS

Symbols of the English alphabet will be replaced by squares with or without points and with or without surrounding lines using the following rule:

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	

OTHER EXAMPLES of MONOALPHABETIC CRYPTOSYSTEMS

Symbols of the English alphabet will be replaced by squares with or without points and with or without surrounding lines using the following rule:

A:	B:	C:
D:	E:	F:
G:	H:	I:

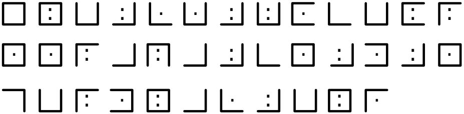
J·	K·	L·
M·	N·	O·
P·	Q·	R·

S	T	U
V	W	X
Y	Z	

For example the plaintext:

WE TALK ABOUT FINNISH SAUNA MANY TIMES LATER

results in the cryptotext:



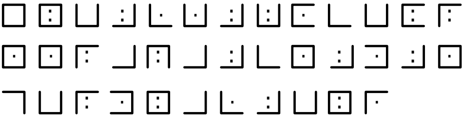
Symbols of the English alphabet will be replaced by squares with or without points and with or without surrounding lines using the following rule:

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

For example the plaintext:

WE TALK ABOUT FINNISH SAUNA MANY TIMES LATER

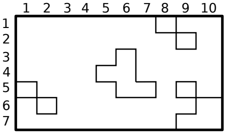
results in the cryptotext:



Garbage in between method: the message (plaintext or cryptotext) is supplemented by "garbage letters".

Richelieu cryptosystem used sheets of card board with holes.

I		L	O	V	E		**Y**	O	U
I		H	A	V	E		Y	**O**	U
D	E	E	P		**U**	N	D	E	R
M	Y		S	**K**	**I**	N		M	Y
L	O	V	E		**L**	**A**	S	**T**	S
F	**O**	R	E	V	E	R		I	**N**
H	Y	P	E	R	S	P	A	**C**	**E**



EXTREME CASES for FREQUENCY ANALYSIS

In 1969 Georges Perec published, in France,

La Disparition

In 1969 Georges Perec published, in France,

La Disparition

a 200 pages novel

In 1969 Georges Perec published, in France,

La Disparition

a 200 pages novel in which there is no occurence of the letter "e".

In 1969 Georges Perec published, in France,

La Disparition

a 200 pages novel in which there is no occurence of the letter "e".

British translation, due to Gilbert Adair, has appeared in 1994 under the title

A void

Appendix A

The Opening Paragraph of *A Void* by Georges Perec, translated by Gilbert Adair

Today, by radio, and also on giant hoardings, a rabbi, an admiral notorious for his links to masonry, a trio of cardinals, a trio, too, of insignificant politicians (bought and paid for by a rich and corrupt Anglo-Canadian banking corporation), inform us all of how our country now risks dying of starvation. A rumor, that's my initial thought as I switch off my radio, a rumor or possibly a hoax. Propaganda, I murmur anxiously–as though, just by saying so, I might allay my doubts–typical politicians' propaganda. But public opinion gradually absorbs it as a fact. Individuals start strutting around with stout clubs. "Food, glorious food!" is a common cry (occasionally sung to Bart's music), with ordinary hardworking folk harassing officials, both local and national, and cursing capitalists and captains of industry. Cops shrink from going out on night shift. In Mâcon a mob storms a municipal building. In Rocadamour ruffians rob a hangar full of foodstuffs, pillaging tons of tuna fish, milk and cocoa, as also a vast quantity of corn–all of it, alas, totally unfit for human consumption. Without fuss or ado, and naturally without any sort of trial, an indignant crowd hangs 26 solicitors on a hastily built scaffold in front of Nancy's law courts (this Nancy is a town, not a woman) and ransacks a local journal, a disgusting right-wing rag that is siding against it. Up and down this land of ours looting has brought docks, shops and farms to a virtual standstill.

HOMOPHONIC CRYPTOSYSTEMS

HOMOPHONIC CRYPTOSYSTEMS

Homophonic cryptosystems are natural generalization of monoalphabetic cryptosystems.

HOMOPHONIC CRYPTOSYSTEMS

Homophonic cryptosystems are natural generalization of monoalphabetic cryptosystems.

They are substitution cryptosystems in which each letter is replaced by arbitrarily chosen substitutes from fixed and disjoint sets of substitutes.

HOMOPHONIC CRYPTOSYSTEMS

Homophonic cryptosystems are natural generalization of monoalphabetic cryptosystems.

They are substitution cryptosystems in which each letter is replaced by arbitrarily chosen substitutes from fixed and disjoint sets of substitutes.

The number of substitutes of a letter is usually proportional to the frequency of the letter.

HOMOPHONIC CRYPTOSYSTEMS

Homophonic cryptosystems are natural generalization of monoalphabetic cryptosystems.

They are substitution cryptosystems in which each letter is replaced by arbitrarily chosen substitutes from fixed and disjoint sets of substitutes.

The number of substitutes of a letter is usually proportional to the frequency of the letter.

Though homophonic cryptosystems are not unbreakable, they are much more secure than ordinary monoalphabetic substitution cryptosystems.

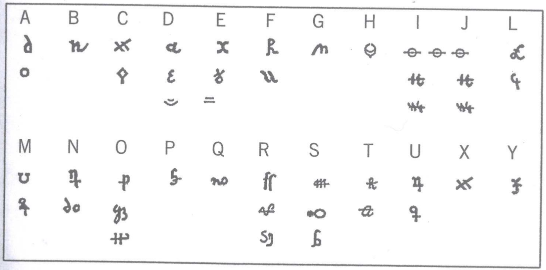
The first known homophonic substitution cipher is from 1401.

Jindřich IV. Francouzský

Homofonní tabulku Jindřicha IV. (viz níže) určitě navrhoval François Viète, oficiální králův kryptograf, luštitel a matematik. Jde o praktickou a účinnou šifru, jakou lze čekat od autora, který zná všechny triky i jejich meze. Většina souhlásek má více variant podle jejich skutečné četnosti. Slovník obsahuje pouhá tři slova.

Tabulka zahrnuje i značkovací symbol:

To stačí k označení všech začátků i konců bezvýznamných úseků, na rozdíl od označování textových částí z Montmorencyho tabulky.



V kódovém seznamu najdeme jen tři slova:

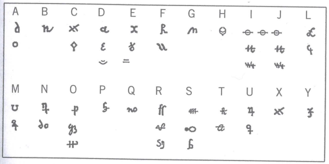
odstavec =   že =   vy =

Jindřich IV. Francouzský

Homofonní tabulku Jindřicha IV. (viz níže) určitě navrhoval François Viète, oficiální králův kryptograf, luštitel a matematik. Jde o praktickou a účinnou šifru, jakou lze čekat od autora, který zná všechny triky i jejich meze. Většina souhlásek má více variant podle jejich skutečné četnosti. Slovník obsahuje pouhá tři slova.

Tabulka zahrnuje i značkovací symbol:

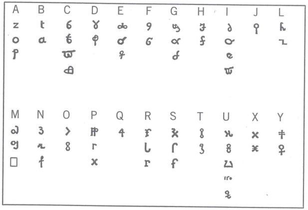
To stačí k označení všech začátků i konců bezvýznamných úseků, na rozdíl od označování textových částí z Montmorencyho tabulky.



V kódovém seznamu najdeme jen tři slova:

odstavec = ože = vy =

Vévoda z Montmorency



Playfair cryptosystem

Invented around 1854 by Ch. Wheatstone.

Playfair cryptosystem

Invented around 1854 by Ch. Wheatstone.

Key – a **Playfair square is defined by a word w of length at most 25. In w repeated letters are then removed, remaining letters of alphabets (except j) are then added and resulting word is divided to form an 5 x 5 array (a Playfair square).**

Playfair cryptosystem

Invented around 1854 by Ch. Wheatstone.

Key – a **Playfair square is defined by a word w of length at most 25. In w repeated letters are then removed, remaining letters of alphabets (except j) are then added and resulting word is divided to form an 5 x 5 array (a Playfair square).**

Encryption: of a pair of letters x, y

1. If x and y are in the same row (column), then they are replaced by the pair of symbols to the right (bellow) them.

Playfair cryptosystem

Invented around 1854 by Ch. Wheatstone.

Key – a **Playfair square is defined by a word w of length at most 25. In w repeated letters are then removed, remaining letters of alphabets (except j) are then added and resulting word is divided to form an 5 x 5 array (a Playfair square).**

Encryption: of a pair of letters x, y

1. If x and y are in the same row (column), then they are replaced by the pair of symbols to the right (bellow) them.
2. If x and y are in different rows and columns they are replaced by symbols in the opposite corners of rectangle created by x and y - the order is important and needs to be agreed on.

Playfair cryptosystem

Invented around 1854 by Ch. Wheatstone.

Key – a **Playfair square is defined by a word w of length at most 25. In w repeated letters are then removed, remaining letters of alphabets (except j) are then added and resulting word is divided to form an 5 x 5 array (a Playfair square).**

Encryption: of a pair of letters x, y

1. If x and y are in the same row (column), then they are replaced by the pair of symbols to the right (bellow) them.
2. If x and y are in different rows and columns they are replaced by symbols in the opposite corners of rectangle created by x and y - the order is important and needs to be agreed on.

Example: PLAYFAIR is encrypted as LCNMNFSC
Playfair was used in World War I by British army.

Playfair square:

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

VIGENERE and AUTOCLAVE cryptosystems

Several of the following polyalphabetic cryptosystems are modification of the CAESAR cryptosystem.

VIGENERE and AUTOCLAVE cryptosystems

Several of the following polyalphabetic cryptosystems are modification of the CAESAR cryptosystem.

Design of cryptosystem: **First step**: A 26×26 table is first designed with the *i*-th row containing all symbols of alphabet, in the cyclic way, starting with *i*-th symbol of the alphabet. This way *i*-th column represent the CAESAR shift $CS(i-1)$ starting with the symbol of the first row.

Second step: For a plaintext w a key k has to be chosen that should be a word of the same length as w.

Encryption: the *i-th* letter of the plaintext - w_i - is encrypted by the letter from the w_i-row and k_i-column of the table.

VIGENERE and AUTOCLAVE cryptosystems

Several of the following polyalphabetic cryptosystems are modification of the CAESAR cryptosystem.

Design of cryptosystem: **First step**: A 26×26 table is first designed with the *i*-th row containing all symbols of alphabet, in the cyclic way, starting with *i*-th symbol of the alphabet. This way *i*-th column represent the CAESAR shift $CS(i-1)$ starting with the symbol of the first row.

Second step: For a plaintext w a key k has to be chosen that should be a word of the same length as w.

Encryption: the *i-th* letter of the plaintext - w_i - is encrypted by the letter from the w_i-row and k_i-column of the table.

VIGENERE cryptosystem is actually a cyclic, key driven, version of the CAESAR cryptosystem.

IMPORTANT EXAMPLES

VIGENERE and AUTOCLAVE cryptosystems

Several of the following polyalphabetic cryptosystems are modification of the CAESAR cryptosystem.

Design of cryptosystem: **First step**: A 26×26 table is first designed with the *i*-th row containing all symbols of alphabet, in the cyclic way, starting with *i*-th symbol of the alphabet. This way *i*-th column represent the CAESAR shift $CS(i-1)$ starting with the symbol of the first row.

Second step: For a plaintext w a key k has to be chosen that should be a word of the same length as w.

Encryption: the *i-th* letter of the plaintext - w_i - is encrypted by the letter from the w_i-row and k_i-column of the table.

VIGENERE cryptosystem is actually a cyclic, key driven, version of the CAESAR cryptosystem.

IMPORTANT EXAMPLES

VIGENERE-key cryptosystem: a short keyword p is chosen and periodically repeated to form the key to be used

VIGENERE and AUTOCLAVE cryptosystems

Several of the following polyalphabetic cryptosystems are modification of the CAESAR cryptosystem.

Design of cryptosystem: **First step**: A 26×26 table is first designed with the *i*-th row containing all symbols of alphabet, in the cyclic way, starting with *i*-th symbol of the alphabet. This way *i*-th column represent the CAESAR shift $CS(i-1)$ starting with the symbol of the first row.

Second step: For a plaintext w a key k has to be chosen that should be a word of the same length as w.

Encryption: the *i-th* letter of the plaintext - w_i - is encrypted by the letter from the w_i-row and k_i-column of the table.

VIGENERE cryptosystem is actually a cyclic, key driven, version of the CAESAR cryptosystem.

IMPORTANT EXAMPLES

VIGENERE-key cryptosystem: a short keyword p is chosen and periodically repeated to form the key to be used

$$k = Prefix_{|w|} p^{oo}$$

VIGENERE and AUTOCLAVE cryptosystems

Several of the following polyalphabetic cryptosystems are modification of the CAESAR cryptosystem.

Design of cryptosystem: **First step**: A 26×26 table is first designed with the *i*-th row containing all symbols of alphabet, in the cyclic way, starting with *i*-th symbol of the alphabet. This way *i*-th column represent the CAESAR shift $CS(i-1)$ starting with the symbol of the first row.

Second step: For a plaintext w a key k has to be chosen that should be a word of the same length as w.

Encryption: the *i-th* letter of the plaintext - w_i - is encrypted by the letter from the w_i-row and k_i-column of the table.

VIGENERE cryptosystem is actually a cyclic, key driven, version of the CAESAR cryptosystem.

IMPORTANT EXAMPLES

VIGENERE-key cryptosystem: a short keyword p is chosen and periodically repeated to form the key to be used

$$k = Prefix_{|w|} p^{oo}$$

AUTOCLAVE-key cryptosystem: a short keyword is chosen and appended by plaintext

POLYALPHABETIC SUBSTITUTION CRYPTOSYSTEMS III

VIGENERE and AUTOCLAVE cryptosystems

Vigenére table:

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

POLYALPHABETIC SUBSTITUTION CRYPTOSYSTEMS III

VIGENERE and AUTOCLAVE cryptosystems

Vigenére table:

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

Keyword: H A M B U R G

Plaintext: I N J E D E M M E N S C H E N G E S I C H T E S T E H T S E I N E G

Vigenere-key:

Autoclave-key:

Vigenere-encrypt..:

Autoclave-encrypt.:

VIGENERE and AUTOCLAVE cryptosystems

Vigenére table:

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

Keyword: H A M B U R G

Plaintext: I N J E D E M M E N S C H E N G E S I C H T E S T E H T S E I N E G

Vigenere-key: H A M B U R G H A M B U R G H A M B U R G H A M B U R G H A M B U R

Autoclave-key:

Vigenere-encrypt..:

Autoclave-encrypt.:

POLYALPHABETIC SUBSTITUTION CRYPTOSYSTEMS III

VIGENERE and AUTOCLAVE cryptosystems

Vigenére table:

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

Keyword: H A M B U R G
Plaintext: I N J E D E M M E N S C H E N G E S I C H T E S T E H T S E I N E G
Vigenere-key: H A M B U R G H A M B U R G H A M B U R G H A M B U R G H A M B U R
Autoclave-key: H A M B U R G I N J E D E M M E N S C H E N G E S I C H T E S T E H
Vigenere-encrypt..: P N V F X V S T E Z T W Y K U G Q T C T N A E E U Y Y Z Z E U O Y X
Autoclave-encrypt.: P N V F X V S U R W W F L Q Z K R K K J L G K W L M J A L I A G I N

COMMENT

- Autoclave-key cipher is also called autokey cipher.

COMMENT

- Autoclave-key cipher is also called autokey cipher.
- So called **running-key cipher** uses very long key that is a passage from a book (for example from Bible).

BLAISE de VIGENERE (1523-1596)

HISTORICAL COMMENT

The encryption method that is commonly called as Vigenere method was actually discovered in 1553 by Giovan Batista Belaso.

VIGÉNERE CRYPTOSYSTEM

- Vigenére work culminated in his *Traicté des Chiffres* - "A treatise on secret writing" in 1586.

VIGÉNERE CRYPTOSYSTEM

- Vigenére work culminated in his *Traicté des Chiffres* - "A treatise on secret writing" in 1586.
- VIGENERE cryptosystem was practically not used for the next 200 years, in spite of its perfection.

VIGÉNERE CRYPTOSYSTEM

- Vigenére work culminated in his *Traicté des Chiffres* - "A treatise on secret writing" in 1586.
- VIGENERE cryptosystem was practically not used for the next 200 years, in spite of its perfection.
- It seems that the reason for ignorance of the VIGENERE cryptosystem was its apparent complexity.

CRYPTANALYSIS of cryptotexts produced by VIGENERE-key cryptosystems

CRYPTANALYSIS of cryptotexts produced by VIGENERE-key cryptosystems

1. Task 1 – to find the length of the keyword

CRYPTANALYSIS of cryptotexts produced by VIGENERE-key cryptosystems

1. Task 1 – to find the length of the keyword

Kasiski's (Prussian officier) method (published in 1862) - invented also by Charles Babbage (1853 - unpublished).

CRYPTANALYSIS of cryptotexts produced by VIGENERE-key cryptosystems

1. Task 1 – to find the length of the keyword

Kasiski's (Prussian officier) method (published in 1862) - invented also by Charles Babbage (1853 - unpublished).

Basic observation: **If a subword of a plaintext is repeated at a distance that is a multiple of the length of the keyword, then the corresponding subwords of the cryptotext have to be the same.**

CRYPTANALYSIS of cryptotexts produced by VIGENERE-key cryptosystems

1. Task 1 – to find the length of the keyword

Kasiski's (Prussian officier) method (published in 1862) - invented also by Charles Babbage (1853 - unpublished).

Basic observation: **If a subword of a plaintext is repeated at a distance that is a multiple of the length of the keyword, then the corresponding subwords of the cryptotext have to be the same.**

Example, cryptotext:

CHRGQPWOEIRULYANDOSHCHRIZKEBUSNOFKYWROPDCHRKGAXBNRHROAKERBKSCHRIWK

CRYPTANALYSIS of cryptotexts produced by VIGENERE-key cryptosystems

1. Task 1 – to find the length of the keyword

Kasiski's (Prussian officier) method (published in 1862) - invented also by Charles Babbage (1853 - unpublished).

Basic observation: **If a subword of a plaintext is repeated at a distance that is a multiple of the length of the keyword, then the corresponding subwords of the cryptotext have to be the same.**

Example, cryptotext:

CHRGQPWOEIRULYANDOSHCHRIZKEBUSNOFKYWROPDCHRKGAXBNRHROAKERBKSCHRIWK

Substring "CHR" occurs in positions 1, 21, 41, 66: expected keyword length is therefore 5.

CRYPTANALYSIS of cryptotexts produced by VIGENERE-key cryptosystems

1. Task 1 – to find the length of the keyword

Kasiski's (Prussian officier) method (published in 1862) - invented also by Charles Babbage (1853 - unpublished).

Basic observation: **If a subword of a plaintext is repeated at a distance that is a multiple of the length of the keyword, then the corresponding subwords of the cryptotext have to be the same.**

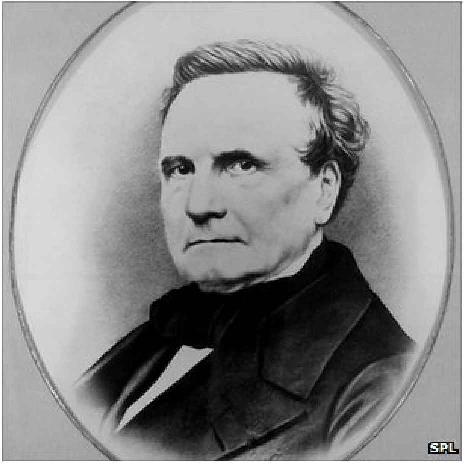
Example, cryptotext:

CHRGQPWOEIRULYANDOSHCHRIZKEBUSNOFKYWROPDCHRKGAXBNRHROAKERBKSCHRIWK

Substring "CHR" occurs in positions 1, 21, 41, 66: expected keyword length is therefore 5.

Method. Determine the greatest common divisor of the distances between identical subwords (of length 3 or more) of the cryptotext.

Charles Babbage (1791-1871)



FRIEDMAN METHOD to DETERMINE KEY LENGTH

Friedman method to determine the length of the keyword:

FRIEDMAN METHOD to DETERMINE KEY LENGTH

Friedman method to determine the length of the keyword: Let n_i be the number of occurrences of the **i**-*th* letter in the cryptotext.

FRIEDMAN METHOD to DETERMINE KEY LENGTH

Friedman method to determine the length of the keyword: Let n_i be the number of occurrences of the **i**-*th* letter in the cryptotext.

Let **L** be the length of the keyword.

FRIEDMAN METHOD to DETERMINE KEY LENGTH

Friedman method to determine the length of the keyword: Let n_i be the number of occurrences of the **i**-*th* letter in the cryptotext.

Let **L** be the length of the keyword.

Let **n** be the length of the cryptotext.

FRIEDMAN METHOD to DETERMINE KEY LENGTH

Friedman method to determine the length of the keyword: Let n_i be the number of occurrences of the **i**-*th* letter in the cryptotext.

Let **L** be the length of the keyword.

Let **n** be the length of the cryptotext.

Then it holds, as shown on next slide:

$$L = \frac{0.027n}{(n-1)I - 0.038n + 0.065}, \; I = \sum_{i=1}^{26} \frac{n_i(n_i - 1)}{n(n-1)}$$

FRIEDMAN METHOD to DETERMINE KEY LENGTH

Friedman method to determine the length of the keyword: Let n_i be the number of occurrences of the **i**-*th* letter in the cryptotext.

Let **L** be the length of the keyword.

Let **n** be the length of the cryptotext.

Then it holds, as shown on next slide:

$$L = \frac{0.027n}{(n-1)I - 0.038n + 0.065}, \quad I = \sum_{i=1}^{26} \frac{n_i(n_i - 1)}{n(n-1)}$$

Once the length of the keyword is found it is easy to determine the key using the frequency analysis method for monoalphabetic cryptosystems.

DERIVATION of the FRIEDMAN METHOD I

DERIVATION of the FRIEDMAN METHOD I

1. Let n be the length of a cryptotext w and o_i be the number of occurrences of the i-th symbol of the alphabet in w.

DERIVATION of the FRIEDMAN METHOD I

1. Let n be the length of a cryptotext w and o_i be the number of occurrences of the i-th symbol of the alphabet in w. The probability that if one selects a pair of symbols from w, then they are the same is

$$I = \frac{\sum_{i=1}^{26} o_i(o_i-1)}{n(n-1)} = \sum_{i=1}^{26} \frac{\binom{o_i}{2}}{\binom{n}{2}}$$

and it is called the index of coincidence.

DERIVATION of the FRIEDMAN METHOD I

1. Let n be the length of a cryptotext w and o_i be the number of occurrences of the i-th symbol of the alphabet in w. The probability that if one selects a pair of symbols from w, then they are the same is

$$I = \frac{\sum_{i=1}^{26} o_i(o_i-1)}{n(n-1)} = \sum_{i=1}^{26} \frac{\binom{o_i}{2}}{\binom{n}{2}}$$

   and it is called the index of coincidence.

2. Let p_i be the probability that a randomly chosen symbol is the i-th symbol of the alphabet.

DERIVATION of the FRIEDMAN METHOD I

1. Let n be the length of a cryptotext w and o_i be the number of occurrences of the i-th symbol of the alphabet in w. The probability that if one selects a pair of symbols from w, then they are the same is

$$I = \frac{\sum_{i=1}^{26} o_i(o_i-1)}{n(n-1)} = \sum_{i=1}^{26} \frac{\binom{o_i}{2}}{\binom{n}{2}}$$

   and it is called the index of coincidence.

2. Let p_i be the probability that a randomly chosen symbol is the i-th symbol of the alphabet. The probability that two randomly chosen symbols are the same is

$$\sum_{i=1}^{26} p_i^2$$

DERIVATION of the FRIEDMAN METHOD I

1. Let n be the length of a cryptotext w and o_i be the number of occurrences of the i-th symbol of the alphabet in w. The probability that if one selects a pair of symbols from w, then they are the same is

$$I = \frac{\sum_{i=1}^{26} o_i(o_i-1)}{n(n-1)} = \sum_{i=1}^{26} \frac{\binom{o_i}{2}}{\binom{n}{2}}$$

and it is called the index of coincidence.

2. Let p_i be the probability that a randomly chosen symbol is the i-th symbol of the alphabet. The probability that two randomly chosen symbols are the same is

$$\sum_{i=1}^{26} p_i^2$$

For an English text one has

$$\sum_{i=1}^{26} p_i^2 = 0.065$$

DERIVATION of the FRIEDMAN METHOD I

1. Let n be the length of a cryptotext w and o_i be the number of occurrences of the i-th symbol of the alphabet in w. The probability that if one selects a pair of symbols from w, then they are the same is

$$I = \frac{\sum_{i=1}^{26} o_i(o_i - 1)}{n(n-1)} = \sum_{i=1}^{26} \frac{\binom{o_i}{2}}{\binom{n}{2}}$$

   and it is called the index of coincidence.

2. Let p_i be the probability that a randomly chosen symbol is the i-th symbol of the alphabet. The probability that two randomly chosen symbols are the same is

$$\sum_{i=1}^{26} p_i^2$$

   For an English text one has

$$\sum_{i=1}^{26} p_i^2 = 0.065$$

   For a randomly chosen text:

$$\sum_{i=1}^{26} p_i^2 = \sum_{i=1}^{26} \frac{1}{26^2} = 0.038$$

DERIVATION of the FRIEDMAN METHOD I

1. Let n be the length of a cryptotext w and o_i be the number of occurrences of the i-th symbol of the alphabet in w. The probability that if one selects a pair of symbols from w, then they are the same is

$$I = \frac{\sum_{i=1}^{26} o_i(o_i - 1)}{n(n-1)} = \sum_{i=1}^{26} \frac{\binom{o_i}{2}}{\binom{n}{2}}$$

and it is called the index of coincidence.

2. Let p_i be the probability that a randomly chosen symbol is the i-th symbol of the alphabet. The probability that two randomly chosen symbols are the same is

$$\sum_{i=1}^{26} p_i^2$$

For an English text one has

$$\sum_{i=1}^{26} p_i^2 = 0.065$$

For a randomly chosen text:

$$\sum_{i=1}^{26} p_i^2 = \sum_{i=1}^{26} \frac{1}{26^2} = 0.038$$

In addition it holds:

$$I = \sum_{i=1}^{26} p_i^2$$

DERIVATION of the FRIEDMAN METHOD - II.

DERIVATION of the FRIEDMAN METHOD - II.

Assume that a cryptotext is writen into L columns headed by the letters of the keyword

key letters	S_1	S_2	S_3	$\dots$	S_L
	x_1	x_2	x_3	$\dots$	x_L
	x_{L+1}	x_{L+2}	x_{L+3}		x_{2L}
	x_{2L+1}	x_{2L+2}	x_{2L+3}	$\dots$	x_{3L}
	.	.	.		.

DERIVATION of the FRIEDMAN METHOD - II.

Assume that a cryptotext is writen into L columns headed by the letters of the keyword

key letters	S_1	S_2	S_3	...	S_L
	x_1	x_2	x_3	...	x_L
	x_{L+1}	x_{L+2}	x_{L+3}		x_{2L}
	x_{2L+1}	x_{2L+2}	x_{2L+3}	...	x_{3L}
	.	.	.		.

First observation Each column is obtained using the CAESAR cryptosystem.

DERIVATION of the FRIEDMAN METHOD - II.

Assume that a cryptotext is writen into L columns headed by the letters of the keyword

key letters	S_1	S_2	S_3	$\dots$	S_L
	x_1	x_2	x_3	$\dots$	x_L
	x_{L+1}	x_{L+2}	x_{L+3}		x_{2L}
	x_{2L+1}	x_{2L+2}	x_{2L+3}	$\dots$	x_{3L}
	.	.	.		.

First observation Each column is obtained using the CAESAR cryptosystem.
Probability that two randomly chosen letters are the same in

- the same column is 0.065.

DERIVATION of the FRIEDMAN METHOD - II.

Assume that a cryptotext is writen into L columns headed by the letters of the keyword

key letters	S_1	S_2	S_3	$\dots$	S_L
	x_1	x_2	x_3	$\dots$	x_L
	x_{L+1}	x_{L+2}	x_{L+3}		x_{2L}
	x_{2L+1}	x_{2L+2}	x_{2L+3}	$\dots$	x_{3L}
	.	.	.		.

First observation Each column is obtained using the CAESAR cryptosystem.
Probability that two randomly chosen letters are the same in

- the same column is 0.065.
- different columns is 0.038.

DERIVATION of the FRIEDMAN METHOD - II.

Assume that a cryptotext is writen into L columns headed by the letters of the keyword

key letters	S_1	S_2	S_3	...	S_L
	x_1	x_2	x_3	...	x_L
	x_{L+1}	x_{L+2}	x_{L+3}		x_{2L}
	x_{2L+1}	x_{2L+2}	x_{2L+3}	...	x_{3L}
	.	.	.		.

First observation Each column is obtained using the CAESAR cryptosystem.
Probability that two randomly chosen letters are the same in

- the same column is 0.065.
- different columns is 0.038.

The number of pairs of letters in the same column: $L \cdot \frac{1}{2} \cdot \frac{n}{L}(\frac{n}{L} - 1) = \frac{n(n-L)}{2L}$

DERIVATION of the FRIEDMAN METHOD - II.

Assume that a cryptotext is writen into L columns headed by the letters of the keyword

key letters	S_1	S_2	S_3	$\dots$	S_L
	x_1	x_2	x_3	$\dots$	x_L
	x_{L+1}	x_{L+2}	x_{L+3}		x_{2L}
	x_{2L+1}	x_{2L+2}	x_{2L+3}	$\dots$	x_{3L}
	.	.	.		.

First observation Each column is obtained using the CAESAR cryptosystem.
Probability that two randomly chosen letters are the same in

- the same column is 0.065.
- different columns is 0.038.

The number of pairs of letters in the same column: $L \cdot \frac{1}{2} \cdot \frac{n}{L}(\frac{n}{L} - 1) = \frac{n(n-L)}{2L}$

The number of pairs of letters in different columns: $\frac{L(L-1)}{2} \cdot \frac{n^2}{L^2} = \frac{n^2(L-1)}{2L}$

The expected number A of pairs of equals letters is $A = \frac{n(n-L)}{2L} \cdot 0.065 + \frac{n^2(L-1)}{2L} \cdot 0.038$

DERIVATION of the FRIEDMAN METHOD - II.

Assume that a cryptotext is writen into L columns headed by the letters of the keyword

key letters	S_1	S_2	S_3	...	S_L
	x_1	x_2	x_3	...	x_L
	x_{L+1}	x_{L+2}	x_{L+3}		x_{2L}
	x_{2L+1}	x_{2L+2}	x_{2L+3}	...	x_{3L}
	.	.	.		.

First observation Each column is obtained using the CAESAR cryptosystem.
Probability that two randomly chosen letters are the same in

- the same column is 0.065.
- different columns is 0.038.

The number of pairs of letters in the same column: $L \cdot \frac{1}{2} \cdot \frac{n}{L}(\frac{n}{L} - 1) = \frac{n(n-L)}{2L}$

The number of pairs of letters in different columns: $\frac{L(L-1)}{2} \cdot \frac{n^2}{L^2} = \frac{n^2(L-1)}{2L}$

The expected number A of pairs of equals letters is $A = \frac{n(n-L)}{2L} \cdot 0.065 + \frac{n^2(L-1)}{2L} \cdot 0.038$

Since $I = \frac{A}{\frac{n(n-1)}{2}} = \frac{1}{L(n-1)}[0.027n + L(0.038n - 0.065)]$

one gets the formula for L from one of the previous slides.

BREAKING VIGENER CRYPTOSYSTEM

BREAKING VIGENER CRYPTOSYSTEM

Kasiski method and the index of coincidence can be used in the following way to break a VIGENERE cryptosystem - basic algorithm.

BREAKING VIGENER CRYPTOSYSTEM

Kasiski method and the index of coincidence can be used in the following way to break a VIGENERE cryptosystem - basic algorithm.

for all guesses of the length *m* of the key
(obtained using Kasiski method) **do**

BREAKING VIGENER CRYPTOSYSTEM

Kasiski method and the index of coincidence can be used in the following way to break a VIGENERE cryptosystem - basic algorithm.

for all guesses of the length m of the key
(obtained using Kasiski method) **do**
    write cryptotext in an array with m columns - row by row;

BREAKING VIGENER CRYPTOSYSTEM

Kasiski method and the index of coincidence can be used in the following way to break a VIGENERE cryptosystem - basic algorithm.

for all guesses of the length *m* of the key
(obtained using Kasiski method) **do**
    write cryptotext in an array with *m* columns - row by row;
    check if index of coincidence of each column is high;

BREAKING VIGENER CRYPTOSYSTEM

Kasiski method and the index of coincidence can be used in the following way to break a VIGENERE cryptosystem - basic algorithm.

for all guesses of the length *m* of the key
(obtained using Kasiski method) **do**
  write cryptotext in an array with *m* columns - row by row;
  check if index of coincidence of each column is high;
  if yes you have the length of key;

BREAKING VIGENER CRYPTOSYSTEM

Kasiski method and the index of coincidence can be used in the following way to break a VIGENERE cryptosystem - basic algorithm.

for all guesses of the length m of the key
(obtained using Kasiski method) **do**
  write cryptotext in an array with m columns - row by row;
  check if index of coincidence of each column is high;
  if yes you have the length of key;

to decode columns use decoding method for Caesar

ONE-TIME PAD CRYPTOSYSTEM – Vernam's cipher

ONE-TIME PAD CRYPTOSYSTEM – Vernam's cipher

Binary case:

plaintext	w	are all binary words of the same length
key	k	
cryptotext	c	

ONE-TIME PAD CRYPTOSYSTEM – Vernam's cipher

Binary case:

$$\left.\begin{array}{ll} \text{plaintext} & w \\ \text{key} & k \\ \text{cryptotext} & c \end{array}\right\} \text{are all binary words of the same length}$$

Encryption: $c = w \oplus k$

Decryption: $w = c \oplus k$

ONE-TIME PAD CRYPTOSYSTEM – Vernam's cipher

Binary case:

plaintext	w	are all binary words of the same length
key	k	
cryptotext	c	

Encryption: $c = w \oplus k$

Decryption: $w = c \oplus k$

Example:

$$w = 101101011$$
$$k = 011011010$$
$$c = 110110001$$

ONE-TIME PAD CRYPTOSYSTEM – Vernam's cipher

Binary case:

plaintext	w	are all binary words of the same length
key	k	
cryptotext	c	

Encryption: $c = w \oplus k$

Decryption: $w = c \oplus k$

Example:

$$w = 101101011$$
$$k = 011011010$$
$$c = 110110001$$

What happens if the same key is used twice or 3 times for encryption?

ONE-TIME PAD CRYPTOSYSTEM – Vernam's cipher

Binary case:

$$\left.\begin{array}{ll} \text{plaintext} & w \\ \text{key} & k \\ \text{cryptotext} & c \end{array}\right\} \text{are all binary words of the same length}$$

Encryption: $c = w \oplus k$

Decryption: $w = c \oplus k$

Example:

$$w = 101101011$$
$$k = 011011010$$
$$c = 110110001$$

What happens if the same key is used twice or 3 times for encryption?

$$\text{If} \quad c_1 = w_1 \oplus k, c_2 = w_2 \oplus k, c_3 = w_3 \oplus k$$

then

ONE-TIME PAD CRYPTOSYSTEM – Vernam's cipher

Binary case:

$$\left.\begin{array}{ll} \text{plaintext} & w \\ \text{key} & k \\ \text{cryptotext} & c \end{array}\right\} \text{ are all binary words of the same length}$$

Encryption: $c = w \oplus k$

Decryption: $w = c \oplus k$

Example:

$$w = 101101011$$
$$k = 011011010$$
$$c = 110110001$$

What happens if the same key is used twice or 3 times for encryption?

$$\text{If} \quad c_1 = w_1 \oplus k, c_2 = w_2 \oplus k, c_3 = w_3 \oplus k$$

then

$$c_1 \oplus c_2 = w_1 \oplus w_2$$
$$c_1 \oplus c_3 = w_1 \oplus w_3$$
$$c_2 \oplus c_3 = w_2 \oplus w_3$$

ONE-TIME PAD CRYPTOSYSTEM – Vernam's cipher

Binary case:

plaintext	w	are all binary words of the same length
key	k	
cryptotext	c	

Encryption: $c = w \oplus k$

Decryption: $w = c \oplus k$

Example:

$$w = 101101011$$
$$k = 011011010$$
$$c = 110110001$$

What happens if the same key is used twice or 3 times for encryption?

If $c_1 = w_1 \oplus k, c_2 = w_2 \oplus k, c_3 = w_3 \oplus k$

then

$$c_1 \oplus c_2 = w_1 \oplus w_2$$
$$c_1 \oplus c_3 = w_1 \oplus w_3$$
$$c_2 \oplus c_3 = w_2 \oplus w_3$$

NEVER USE ONE-TIME PAD TWICE WITH THE SAME KEY

NEVER USE ONE-TIME PAD TWICE WITH THE SAME KEY

The reuse of keys by Soviet Union spies (due to the maanufacturer's accidental duplication of one-time-pad pages) enabled US cryptanalysts to unmask the atomic spy Klaus Fuchs in 1949.

PERFECT SECRET-KEY CRYPTOSYSTEMS- I.

PERFECT SECRET-KEY CRYPTOSYSTEMS- I.

By Shannon a cryptosystem is secure if *a posterior* distribution of the plaintext P after we know the cryptotext C is equal to the *a priory* distribution of the plaintext.

PERFECT SECRET-KEY CRYPTOSYSTEMS- I.

By Shannon a cryptosystem is secure if *a posterior* distribution of the plaintext P after we know the cryptotext C is equal to the *a priory* distribution of the plaintext.

Formally, for all pairs plaintext p and cryptotext c such that $Prob[C = c] \neq 0$ it holds that

$$\mathrm{Prob}[P = p|C = c] = Prob[P = p].$$

PERFECT SECRET-KEY CRYPTOSYSTEMS- I.

By Shannon a cryptosystem is secure if *a posterior* distribution of the plaintext P after we know the cryptotext C is equal to the *a priory* distribution of the plaintext.

Formally, for all pairs plaintext p and cryptotext c such that $Prob[C = c] \neq 0$ it holds that

$$\text{Prob}[P = p|C = c] = Prob[P = p].$$

Example ONE-TIME PAD cryptosystem is perfectly secure because for any pair c, p there exists a key k such that

$$c = k \oplus p.$$

PERFECT SECRECY of ONE-TIME PAD

One-time pad cryptosystem is **perfectly secure** because

PERFECT SECRECY of ONE-TIME PAD

One-time pad cryptosystem is **perfectly secure** because

For any cryptotext

$$c = c_1 c_2 \dots c_n$$

PERFECT SECRECY of ONE-TIME PAD

One-time pad cryptosystem is **perfectly secure** because

For any cryptotext

$$c = c_1 c_2 \dots c_n$$

and any plaintext

$$p = p_1 p_2 \dots p_n$$

PERFECT SECRECY of ONE-TIME PAD

One-time pad cryptosystem is **perfectly secure** because

For any cryptotext

$$c = c_1 c_2 \dots c_n$$

and any plaintext

$$p = p_1 p_2 \dots p_n$$

there exists a key (and all keys were chosen with the same probability)

$$k = k_1 k_2 \dots k_n = \mathbf{p} \oplus \mathbf{c}$$

such that

$$c = p \oplus k$$

PERFECT SECRECY of ONE-TIME PAD

One-time pad cryptosystem is **perfectly secure** because

For any cryptotext

$$c = c_1 c_2 \dots c_n$$

and any plaintext

$$p = p_1 p_2 \dots p_n$$

there exists a key (and all keys were chosen with the same probability)

$$k = k_1 k_2 \dots k_n = \mathbf{p} \oplus \mathbf{c}$$

such that

$$c = p \oplus k$$

Did we gain something? The problem of secure communication of the plaintext got transformed to the problem of secure communication of the key of the same length.

PERFECT SECRECY of ONE-TIME PAD

One-time pad cryptosystem is **perfectly secure** because

For any cryptotext

$$c = c_1 c_2 \dots c_n$$

and any plaintext

$$p = p_1 p_2 \dots p_n$$

there exists a key (and all keys were chosen with the same probability)

$$k = k_1 k_2 \dots k_n = \mathbf{p} \oplus \mathbf{c}$$

such that

$$c = p \oplus k$$

Did we gain something? The problem of secure communication of the plaintext got transformed to the problem of secure communication of the key of the same length.

Yes:

1. ONE-TIME PAD cryptosystem is used in critical applications
2. It suggests an idea how to construct practically secure cryptosystems.

PERFECT SECRECY of ONE-TIME PAD

One-time pad cryptosystem is **perfectly secure** because

For any cryptotext

$$c = c_1 c_2 \dots c_n$$

and any plaintext

$$p = p_1 p_2 \dots p_n$$

there exists a key (and all keys were chosen with the same probability)

$$k = k_1 k_2 \dots k_n = \mathbf{p} \oplus \mathbf{c}$$

such that

$$c = p \oplus k$$

Did we gain something? The problem of secure communication of the plaintext got transformed to the problem of secure communication of the key of the same length.

Yes:

1. ONE-TIME PAD cryptosystem is used in critical applications
2. It suggests an idea how to construct practically secure cryptosystems.

IDEA: Find a simple way to generate almost perfectly random key shared by both communicating parties and make them to use this key for one-time pad encoding and decoding!!!!

PERFECT SECRECY of ONE-TIME PAD ONCE MORE

For

every cryptotext c

PERFECT SECRECY of ONE-TIME PAD ONCE MORE

For

every cryptotext c

every element p of the set of plaintexts has the same probability

that p was the plaintext the encryption of which provided c as the cryptotext.

CURRENT ROLE of SUBSTITUTION SYSTEMS

CURRENT ROLE of SUBSTITUTION SYSTEMS

- Substitution ciphers alone are no longer of use.

CURRENT ROLE of SUBSTITUTION SYSTEMS

- Substitution ciphers alone are no longer of use.
- They can be used in a combination with other ciphers as product ciphers.

CURRENT ROLE of SUBSTITUTION SYSTEMS

- Substitution ciphers alone are no longer of use.
- They can be used in a combination with other ciphers as product ciphers.
- However, from a sufficiently abstract perspective, modern bit-oriented block ciphers (DES, AES,...) can be viewed as substitution ciphers on enormously large binary alphabets.

CURRENT ROLE of SUBSTITUTION SYSTEMS

- Substitution ciphers alone are no longer of use.
- They can be used in a combination with other ciphers as product ciphers.
- However, from a sufficiently abstract perspective, modern bit-oriented block ciphers (DES, AES,...) can be viewed as substitution ciphers on enormously large binary alphabets.
- Moreover, modern block ciphers often include smaller substitution tables, called S-boxes.

TRANSPOSITION CRYPTOSYSTEMS

TRANSPOSITION CRYPTOSYSTEMS

The basic idea is very simple: permute the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

TRANSPOSITION CRYPTOSYSTEMS

The basic idea is very simple: permute the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

One idea: choose *n*, write plaintext into rows, with *n* symbols in each row and then read it by columns to get cryptotext.

Example

I	N	J	E	D	E	M	M	E	N
S	C	H	E	N	G	E	S	I	C
H	T	E	S	T	E	H	T	S	E
I	N	E	G	E	S	C	H	I	C
H	T	E	T	O	J	E	O	N	O

TRANSPOSITION CRYPTOSYSTEMS

The basic idea is very simple: permute the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

One idea: choose n, write plaintext into rows, with n symbols in each row and then read it by columns to get cryptotext.

Example

I	N	J	E	D	E	M	M	E	N
S	C	H	E	N	G	E	S	I	C
H	T	E	S	T	E	H	T	S	E
I	N	E	G	E	S	C	H	I	C
H	T	E	T	O	J	E	O	N	O

Cryptotexts obtained by transpositions, called anagrams, were popular among scientists of 17th century. They were used also to encrypt scientific findings.

TRANSPOSITION CRYPTOSYSTEMS

The basic idea is very simple: permute the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

One idea: choose *n*, write plaintext into rows, with *n* symbols in each row and then read it by columns to get cryptotext.

Example

I	N	J	E	D	E	M	M	E	N
S	C	H	E	N	G	E	S	I	C
H	T	E	S	T	E	H	T	S	E
I	N	E	G	E	S	C	H	I	C
H	T	E	T	O	J	E	O	N	O

Cryptotexts obtained by transpositions, called anagrams, were popular among scientists of 17th century. They were used also to encrypt scientific findings.

Newton wrote to Leibniz

TRANSPOSITION CRYPTOSYSTEMS

The basic idea is very simple: permute the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

One idea: choose n, write plaintext into rows, with n symbols in each row and then read it by columns to get cryptotext.

Example

I	N	J	E	D	E	M	M	E	N
S	C	H	E	N	G	E	S	I	C
H	T	E	S	T	E	H	T	S	E
I	N	E	G	E	S	C	H	I	C
H	T	E	T	O	J	E	O	N	O

Cryptotexts obtained by transpositions, called anagrams, were popular among scientists of 17th century. They were used also to encrypt scientific findings.

Newton wrote to Leibniz

$$a^7 c^2 d^2 e^{14} f^2 i^7 l^3 m^1 n^8 o^4 q^3 r^2 s^4 t^8 v^{12} x^1$$

TRANSPOSITION CRYPTOSYSTEMS

The basic idea is very simple: permute the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

One idea: choose n, write plaintext into rows, with n symbols in each row and then read it by columns to get cryptotext.

Example

I	N	J	E	D	E	M	M	E	N
S	C	H	E	N	G	E	S	I	C
H	T	E	S	T	E	H	T	S	E
I	N	E	G	E	S	C	H	I	C
H	T	E	T	O	J	E	O	N	O

Cryptotexts obtained by transpositions, called anagrams, were popular among scientists of 17th century. They were used also to encrypt scientific findings.

Newton wrote to Leibniz

$$a^7c^2d^2e^{14}f^2i^7l^3m^1n^8o^4q^3r^2s^4t^8v^{12}x^1$$

what stands for: "data aequatione quodcumque fluentes quantitates involvente, fluxiones invenire et vice versa"

TRANSPOSITION CRYPTOSYSTEMS

The basic idea is very simple: permute the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

One idea: choose n, write plaintext into rows, with n symbols in each row and then read it by columns to get cryptotext.

Example

I	N	J	E	D	E	M	M	E	N
S	C	H	E	N	G	E	S	I	C
H	T	E	S	T	E	H	T	S	E
I	N	E	G	E	S	C	H	I	C
H	T	E	T	O	J	E	O	N	O

Cryptotexts obtained by transpositions, called anagrams, were popular among scientists of 17th century. They were used also to encrypt scientific findings.

Newton wrote to Leibniz

$$a^7 c^2 d^2 e^{14} f^2 i^7 l^3 m^1 n^8 o^4 q^3 r^2 s^4 t^8 v^{12} x^1$$

what stands for: "data aequatione quodcumque fluentes quantitates involvente, fluxiones invenire et vice versa"

Example

TRANSPOSITION CRYPTOSYSTEMS

The basic idea is very simple: permute the plaintext to get the cryptotext. Less clear it is how to specify and perform efficiently permutations.

One idea: choose n, write plaintext into rows, with n symbols in each row and then read it by columns to get cryptotext.

Example

I	N	J	E	D	E	M	M	E	N
S	C	H	E	N	G	E	S	I	C
H	T	E	S	T	E	H	T	S	E
I	N	E	G	E	S	C	H	I	C
H	T	E	T	O	J	E	O	N	O

Cryptotexts obtained by transpositions, called anagrams, were popular among scientists of 17th century. They were used also to encrypt scientific findings.

Newton wrote to Leibniz

$$a^7c^2d^2e^{14}f^2i^7l^3m^1n^8o^4q^3r^2s^4t^8v^{12}x^1$$

what stands for: "data aequatione quodcumque fluentes quantitates involvente, fluxiones invenire et vice versa"

Example $a^2cdef^3g^2i^2jkmn^3o^5prs^2t^2u^3z$

Solution: ??

KEYWORD CAESAR CRYPTOSYSTEM

KEYWORD CAESAR CRYPTOSYSTEM

This will be an example showing that cryptanalysis often require qualified guessing.

KEYWORD CAESAR CRYPTOSYSTEM

This will be an example showing that cryptanalysis often require qualified guessing.

Keyword Caesar cryptosystem is given by choosing an integer $0 < k < 25$ and a string, called keyword, of length at most 25 with all letters different.

KEYWORD CAESAR CRYPTOSYSTEM

This will be an example showing that cryptanalysis often require qualified guessing.
Keyword Caesar cryptosystem is given by choosing an integer $0 < k < 25$ and a string, called keyword, of length at most 25 with all letters different.

The keyword is then written bellow the English alphabet letters, beginning with the k-symbol, and the remaining letters are written in the alphabetic order and cyclically after the keyword.

KEYWORD CAESAR CRYPTOSYSTEM

This will be an example showing that cryptanalysis often require qualified guessing.
Keyword Caesar cryptosystem is given by choosing an integer $0 < k < 25$ and a string, called keyword, of length at most 25 with all letters different.

The keyword is then written bellow the English alphabet letters, beginning with the k-symbol, and the remaining letters are written in the alphabetic order and cyclically after the keyword.

Example: keyword: HOW MANY ELKS, $k = 8$

0								8																	
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
P	Q	R	T	U	V	X	Z	H	O	W	M	A	N	Y	E	L	K	S	B	C	D	F	G	I	J

KEYWORD CAESAR - Example I

Example Decrypt the following cryptotext encrypted using the KEYWORD CAESAR and determine the keyword and k

```
T  IVD  ZCRTIC  FQNIQ  TU  TF
Q  XAVFCZ  FEQXC  PCQUCZ  WK
Q  FUVBC  FNRRTXTCIUAK  WTY
DTUP  MCFECXU  UV  UPC  BVANHC
VR  UPC  FEQXC  UPC  FUVBC
XVIUQTIF  FUVICF  NFNQAAK
VI  UPC  UVE  UV  UQGC  Q  FQNIQ
WQUP  TU  TF  QAFV  ICXCFFQMK
UPQU  UPC  FUVBC  TF  EMVECMAK
PCQUCZ  QIZ  UPQU  KVN  PQBC
UPC  RQXTATUK  VR  UPMVDTIY
DQUCM  VI  UPC  FUVICF
```

Step 1. Make the frequency counts:

	Number
U	32
C	31
Q	23
F	22
V	20
P	15
T	15
I	14
A	8
	180=74.69%

	Number
X	8
K	7
N	7
E	6
M	6
R	6
B	5
Z	5
D	4
	54=22.41%

	Number
W	3
Y	2
G	1
H	1
J	0
L	0
O	0
S	0
	7=2.90%

KEYWORD CAESAR - Example II

Step 1. Make the frequency counts:

	Number		Number		Number
U	32	X	8	W	3
C	31	K	7	Y	2
Q	23	N	7	G	1
F	22	E	6	H	1
V	20	M	6	J	0
P	15	R	6	L	0
T	15	B	5	O	0
I	14	Z	5	S	0
A	8	D	4		
	180=74.69%		54=22.41%		7=2.90%

Step 2. Cryptotext contains two one-letter words T and Q.

KEYWORD CAESAR - Example II

Step 1. Make the frequency counts:

	Number		Number		Number
U	32	X	8	W	3
C	31	K	7	Y	2
Q	23	N	7	G	1
F	22	E	6	H	1
V	20	M	6	J	0
P	15	R	6	L	0
T	15	B	5	O	0
I	14	Z	5	S	0
A	8	D	4		
	180=74.69%		54=22.41%		7=2.90%

Step 2. Cryptotext contains two one-letter words T and Q. They must be A and I. Since T occurs once and Q three times it is likely that T is I and Q is A.

KEYWORD CAESAR - Example II

Step 1. Make the frequency counts:

	Number
U	32
C	31
Q	23
F	22
V	20
P	15
T	15
I	14
A	8
	180=74.69%

	Number
X	8
K	7
N	7
E	6
M	6
R	6
B	5
Z	5
D	4
	54=22.41%

	Number
W	3
Y	2
G	1
H	1
J	0
L	0
O	0
S	0
	7=2.90%

Step 2. Cryptotext contains two one-letter words T and Q.They must be A and I. Since T occurs once and Q three times it is likely that T is I and Q is A.

The three letter word UPC occurs 7 times and all other 3-letter words occur only once.

KEYWORD CAESAR - Example II

Step 1. Make the frequency counts:

	Number		Number		Number
U	32	X	8	W	3
C	31	K	7	Y	2
Q	23	N	7	G	1
F	22	E	6	H	1
V	20	M	6	J	0
P	15	R	6	L	0
T	15	B	5	O	0
I	14	Z	5	S	0
A	8	D	4		
	180=74.69%		54=22.41%		7=2.90%

Step 2. Cryptotext contains two one-letter words T and Q.They must be A and I. Since T occurs once and Q three times it is likely that T is I and Q is A.

The three letter word UPC occurs 7 times and all other 3-letter words occur only once. Hence

UPC is likely to be THE.

Let us now decrypt the remaining letters in the high frequency group: F,V,I

From the words TU, TF $\Rightarrow$ F=S
From UV $\Rightarrow$ V=O
From VI $\Rightarrow$ I=N

CONTINUATION

So we have: T=I, Q=A, U=T, P=H, C=E, F=S, V=O, I=N and now in

```
T   I V D   Z C R T I C   F Q N I Q   T U   T F
Q   X A V F C Z   F E Q X C   P C Q U C Z   W K
Q   F U V B C   F N R R T X T C I U A K   W T Y
D T U P   M C F E C X U   U V   U P C   B V A N H C
V R   U P C   F E Q X C   U P C   F U V B C
X V I U Q T I F   F U V I C F   N F N Q A A K
V I   U P C   U V E   U V   U Q G C   Q   F Q N I Q
W Q U P   T U   T F   Q A F V   I C X C F F Q M K
U P Q U   U P C   F U V B C   T F   E M V E C M A K
P C Q U C Z   Q I Z   U P Q U   K V N   P Q B C
U P C   R Q X T A T U K   V R   U P M V D T I Y
D Q U C M   V I   U P C   F U V I C F
```

we have several words with only one unknown letter what leads to another guesses and the table:

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
L	V	E	W	P	S	K	M	N	?	Y	?	R	U	?	H	A	F	?	I	T	O	B	C	G	D

This leads to the keyword **CRYPTOGRAPHY GIVES ME FUN** and $k = 4$ -

SHANNON's CONTRIBUTIONS to UNDERSTANDING CIPHERS

SHANNON's CONTRIBUTIONS to UNDERSTANDING CIPHERS

- For understanding the quality of secret key ciphers of large importance was Clause Shannon's paper **A Communication Theory of Secrecy systems**.

SHANNON's CONTRIBUTIONS to UNDERSTANDING CIPHERS

- For understanding the quality of secret key ciphers of large importance was Clause Shannon's paper **A Communication Theory of Secrecy systems**.
- Shannon introduced several advance mathematical technique to scientific cryptography.

SHANNON's CONTRIBUTIONS to UNDERSTANDING CIPHERS

- For understanding the quality of secret key ciphers of large importance was Clause Shannon's paper **A Communication Theory of Secrecy systems**.
- Shannon introduced several advance mathematical technique to scientific cryptography.
- Shannon demonstrated several important features of the statical nature of natural languages that makes solution to many problems of ciphers very straightforward.

SHANNON's CONTRIBUTIONS to UNDERSTANDING CIPHERS

- For understanding the quality of secret key ciphers of large importance was Clause Shannon's paper **A Communication Theory of Secrecy systems**.
- Shannon introduced several advance mathematical technique to scientific cryptography.
- Shannon demonstrated several important features of the statical nature of natural languages that makes solution to many problems of ciphers very straightforward.
- One of the main contribution of the above Shannon's paper was the development of a measure, called **unicity distane**,

SHANNON's CONTRIBUTIONS to UNDERSTANDING CIPHERS

- For understanding the quality of secret key ciphers of large importance was Clause Shannon's paper **A Communication Theory of Secrecy systems**.
- Shannon introduced several advance mathematical technique to scientific cryptography.
- Shannon demonstrated several important features of the statical nature of natural languages that makes solution to many problems of ciphers very straightforward.
- One of the main contribution of the above Shannon's paper was the development of a measure, called **unicity distane**, of the cryptohgraphic strength of the ciphers encoding messages of natural languages.

UNICITY DISTANCE - MOTIVATION - INFORMALLY

UNICITY DISTANCE - MOTIVATION - INFORMALLY

The unicity distance of a cipher encrypting natural language plaintexts is the minimum of cryptotexts required for computationally unlimited adversaries to decrypt cryptotext uniquely (to recover uniquely the key that was used).

UNICITY DISTANCE - MOTIVATION - INFORMALLY

The unicity distance of a cipher encrypting natural language plaintexts is the minimum of cryptotexts required for computationally unlimited adversaries to decrypt cryptotext uniquely (to recover uniquely the key that was used).

- **Example 1**: Let **WNAIW** be the cryptotext obtained by encoding an English word by Vigenere key cipher with the key of the length 5.

UNICITY DISTANCE - MOTIVATION - INFORMALLY

The unicity distance of a cipher encrypting natural language plaintexts is the minimum of cryptotexts required for computationally unlimited adversaries to decrypt cryptotext uniquely (to recover uniquely the key that was used).

- **Example 1**: Let **WNAIW** be the cryptotext obtained by encoding an English word by Vigenere key cipher with the key of the length 5. Can one determine uniquely the plaintext?

UNICITY DISTANCE - MOTIVATION - INFORMALLY

The unicity distance of a cipher encrypting natural language plaintexts is the minimum of cryptotexts required for computationally unlimited adversaries to decrypt cryptotext uniquely (to recover uniquely the key that was used).

- **Example 1**: Let **WNAIW** be the cryptotext obtained by encoding an English word by Vigenere key cipher with the key of the length 5. Can one determine uniquely the plaintext?
- One can find two fully satisfactory solutions: **RIVER, WATER**

UNICITY DISTANCE - MOTIVATION - INFORMALLY

The unicity distance of a cipher encrypting natural language plaintexts is the minimum of cryptotexts required for computationally unlimited adversaries to decrypt cryptotext uniquely (to recover uniquely the key that was used).

- **Example 1**: Let **WNAIW** be the cryptotext obtained by encoding an English word by Vigenere key cipher with the key of the length 5. Can one determine uniquely the plaintext?
- One can find two fully satisfactory solutions: **RIVER, WATER** and many nonsatisfactory as **KHDOP, SXOOS**, but not the unique plaintext.

UNICITY DISTANCE - MOTIVATION - INFORMALLY

The unicity distance of a cipher encrypting natural language plaintexts is the minimum of cryptotexts required for computationally unlimited adversaries to decrypt cryptotext uniquely (to recover uniquely the key that was used).

- **Example 1**: Let **WNAIW** be the cryptotext obtained by encoding an English word by Vigenere key cipher with the key of the length 5. Can one determine uniquely the plaintext?
- One can find two fully satisfactory solutions: **RIVER, WATER** and many nonsatisfactory as **KHDOP, SXOOS**, but not the unique plaintext.
- **Example 2**: Let cryptotext **FJKFPO** was obtained by encrypting an English text using a monoalphabetic substitution cipher. Can we find the unique plaintext?

UNICITY DISTANCE - MOTIVATION - INFORMALLY

The unicity distance of a cipher encrypting natural language plaintexts is the minimum of cryptotexts required for computationally unlimited adversaries to decrypt cryptotext uniquely (to recover uniquely the key that was used).

- **Example 1**: Let **WNAIW** be the cryptotext obtained by encoding an English word by Vigenere key cipher with the key of the length 5. Can one determine uniquely the plaintext?
- One can find two fully satisfactory solutions: **RIVER, WATER** and many nonsatisfactory as **KHDOP, SXOOS**, but not the unique plaintext.
- **Example 2**: Let cryptotext **FJKFPO** was obtained by encrypting an English text using a monoalphabetic substitution cipher. Can we find the unique plaintext?
- Possible plaintexts are **thatis, ofyour, season, oxford, thatof,....** but there is no way to determine the plaintext uniquely.

UNICITY DISTANCE - BASIC RESULT

UNICITY DISTANCE - BASIC RESULT

The expected unicity distance $U_{C,K,L}$ of a cipher C and a key set K for a plaintext language L can be shown to be:

$$U_{C,K,L} = \frac{H_K}{D_L}$$

where H_K is the entropy of the key space (e.g 128 for 2^{128} equiprobably keys), D_L is the plaintext redundancy in bits per character.

UNICITY DISTANCE - BASIC RESULT

The expected unicity distance $U_{C,K,L}$ of a cipher C and a key set K for a plaintext language L can be shown to be:

$$U_{C,K,L} = \frac{H_K}{D_L}$$

where H_K is the entropy of the key space (e.g 128 for 2^{128} equiprobably keys), D_L is the plaintext redundancy in bits per character.

Redundancy: Each character in English can convey $\lg(26) = 4.7$ bits of information.

However, the average amount of actual information carried per character in a meaningful English text is only about 1.5 bits per character.

UNICITY DISTANCE - BASIC RESULT

The expected unicity distance $U_{C,K,L}$ of a cipher C and a key set K for a plaintext language L can be shown to be:

$$U_{C,K,L} = \frac{H_K}{D_L}$$

where H_K is the entropy of the key space (e.g 128 for 2^{128} equiprobably keys), D_L is the plaintext redundancy in bits per character.

Redundancy: Each character in English can convey $\lg(26) = 4.7$ bits of information.

However, the average amount of actual information carried per character in a meaningful English text is only about 1.5 bits per character.

So the plaintext redundancy is $4.7 - 1.5 = 3.2$.

EXAMPLES

EXAMPLES

Simple monoalphabetic substitution cipher: Number of possible keys is $26! \approx 2^{88.4}$. Assuming that all keys (permutations) are are equally probable we have $H_K = \lg(26!) = 88.4$ bits.

EXAMPLES

Simple monoalphabetic substitution cipher: Number of possible keys is $26! \approx 2^{88.4}$.
Assuming that all keys (permutations) are are equally probable we have $H_K = \lg(26!) = 88.4$ bits.

Since for English text $D_L = 3.2$, we have for the unicity distance

$$U = \frac{88.4}{3.2} = 28$$

Conclusion Given at least 28 characters of the cryptotext it should be possible, at least theoretically, to find unique plaintext (and key).

EXAMPLES

Simple monoalphabetic substitution cipher: Number of possible keys is $26! \approx 2^{88.4}$.
Assuming that all keys (permutations) are are equally probable we have $H_K = \lg(26!) = 88.4$ bits.

Since for English text $D_L = 3.2$, we have for the unicity distance

$$U = \frac{88.4}{3.2} = 28$$

Conclusion Given at least 28 characters of the cryptotext it should be possible, at least theoretically, to find unique plaintext (and key).

Other ciphers:

- **Atbash cipher:** Number of keys: 1; unicity distance: 0 characters

EXAMPLES

Simple monoalphabetic substitution cipher: Number of possible keys is $26! \approx 2^{88.4}$.
Assuming that all keys (permutations) are are equally probable we have $H_K = \lg(26!) = 88.4$ bits.

Since for English text $D_L = 3.2$, we have for the unicity distance

$$U = \frac{88.4}{3.2} = 28$$

Conclusion Given at least 28 characters of the cryptotext it should be possible, at least theoretically, to find unique plaintext (and key).

Other ciphers:

- **Atbash cipher:** Number of keys: 1; unicity distance: 0 characters
- **Ceaser cipher:** Number of keys: 25; unicity distance: 2 characters

EXAMPLES

Simple monoalphabetic substitution cipher: Number of possible keys is $26! \approx 2^{88.4}$.
Assuming that all keys (permutations) are are equally probable we have $H_K = \lg(26!) = 88.4$ bits.

Since for English text $D_L = 3.2$, we have for the unicity distance

$$U = \frac{88.4}{3.2} = 28$$

Conclusion Given at least 28 characters of the cryptotext it should be possible, at least theoretically, to find unique plaintext (and key).

Other ciphers:

- **Atbash cipher:** Number of keys: 1; unicity distance: 0 characters
- **Ceaser cipher:** Number of keys: 25; unicity distance: 2 characters
- **Affine cipher:** Number of keys: 311; unicity distance: 3

EXAMPLES

Simple monoalphabetic substitution cipher: Number of possible keys is $26! \approx 2^{88.4}$.
Assuming that all keys (permutations) are are equally probable we have $H_K = \lg(26!) = 88.4$ bits.

Since for English text $D_L = 3.2$, we have for the unicity distance

$$U = \frac{88.4}{3.2} = 28$$

Conclusion Given at least 28 characters of the cryptotext it should be possible, at least theoretically, to find unique plaintext (and key).

Other ciphers:

- **Atbash cipher:** Number of keys: 1; unicity distance: 0 characters
- **Ceaser cipher:** Number of keys: 25; unicity distance: 2 characters
- **Affine cipher:** Number of keys: 311; unicity distance: 3
- **Playfair cipher:** Number of keys: 25!; unicity distance: 27

COMMENTS

COMMENTS

- Observe that Unicity distance is only a theoretical minimum.

COMMENTS

- Observe that Unicity distance is only a theoretical minimum.
- In general one may need much more characters to reliably break a cipher - say 100 for simple monoalphabetic substitution cipher.

COMMENTS

- Observe that Unicity distance is only a theoretical minimum.
- In general one may need much more characters to reliably break a cipher - say 100 for simple monoalphabetic substitution cipher.
- Unicity distance is a useful theoretical measure, but it does not say much about security of a block cipher when attacked by an adversary with real-world (limited) resources.

COMMENTS

- Observe that Unicity distance is only a theoretical minimum.
- In general one may need much more characters to reliably break a cipher - say 100 for simple monoalphabetic substitution cipher.
- Unicity distance is a useful theoretical measure, but it does not say much about security of a block cipher when attacked by an adversary with real-world (limited) resources.
- Unicity distance is not a measure of how much cryptotext is needed for ctyptanalysis,

COMMENTS

- Observe that Unicity distance is only a theoretical minimum.
- In general one may need much more characters to reliably break a cipher - say 100 for simple monoalphabetic substitution cipher.
- Unicity distance is a useful theoretical measure, but it does not say much about security of a block cipher when attacked by an adversary with real-world (limited) resources.
- Unicity distance is not a measure of how much cryptotext is needed for ctyptanalysis, but how much cryptotext is required for there to be only one reasonable solution for cryptanalysis.

UNICITY DISTANCE of CRYPTOSYSTEMS - INFORMALLY

UNICITY DISTANCE of CRYPTOSYSTEMS - INFORMALLY

Redundancy of natural languages is of the key importance for cryptanalysis.

UNICITY DISTANCE of CRYPTOSYSTEMS - INFORMALLY

Redundancy of natural languages is of the key importance for cryptanalysis.

Would all letters of a 26-symbol alphabet have the same probability, a character would carry $\lg 26 = 4.7$ bits of Information.

UNICITY DISTANCE of CRYPTOSYSTEMS - INFORMALLY

Redundancy of natural languages is of the key importance for cryptanalysis.

Would all letters of a 26-symbol alphabet have the same probability, a character would carry $\lg 26 = 4.7$ bits of Information.

The estimated average amount of information carried per letter in a meaningful English text is 1.5 bits.

UNICITY DISTANCE of CRYPTOSYSTEMS - INFORMALLY

Redundancy of natural languages is of the key importance for cryptanalysis.

Would all letters of a 26-symbol alphabet have the same probability, a character would carry $\lg 26 = 4.7$ bits of Information.

The estimated average amount of information carried per letter in a meaningful English text is 1.5 bits.

The unicity distance of a cryptosystem is the minimum length of the cryptotext required by a computationally unlimited adversary to recover the unique encryption key.

UNICITY DISTANCE of CRYPTOSYSTEMS - INFORMALLY

Redundancy of natural languages is of the key importance for cryptanalysis.

Would all letters of a 26-symbol alphabet have the same probability, a character would carry $\lg 26 = 4.7$ bits of Information.

The estimated average amount of information carried per letter in a meaningful English text is 1.5 bits.

The unicity distance of a cryptosystem is the minimum length of the cryptotext required by a computationally unlimited adversary to recover the unique encryption key.

Empirical evidence indicates that if a simple substitution cryptosystem is applied to a a meaningful English message, then about 25 cryptotext characters are enough for an experienced cryptanalyst to recover the plaintext.

UNICITY DISTANCE - MORE TECHNICALLY

UNICITY DISTANCE - MORE TECHNICALLY

- The unicity distance of a cryptosystem is a number that indicates the amount of cryptotext required in order to uniquely determine the plaintext.

UNICITY DISTANCE - MORE TECHNICALLY

- The unicity distance of a cryptosystem is a number that indicates the amount of cryptotext required in order to uniquely determine the plaintext.
- It is a function of the length of the key used for encryption and the statistical nature of the plaintext language.

UNICITY DISTANCE - MORE TECHNICALLY

- The unicity distance of a cryptosystem is a number that indicates the amount of cryptotext required in order to uniquely determine the plaintext.
- It is a function of the length of the key used for encryption and the statistical nature of the plaintext language.
- Outcome: It can be shown that enough time it is guaranteed that any cipher can be broken given cryptotexts of such total length that unicity distance is 1.

UNICITY DISTANCE - MORE TECHNICALLY

- The unicity distance of a cryptosystem is a number that indicates the amount of cryptotext required in order to uniquely determine the plaintext.
- It is a function of the length of the key used for encryption and the statistical nature of the plaintext language.
- Outcome: It can be shown that enough time it is guaranteed that any cipher can be broken given cryptotexts of such total length that unicity distance is 1.
- Formally, the unicity distance is given by

$$\frac{H(K)}{|M| - H(M)}$$

  where

  - $H(k)$ is information content of the key.

UNICITY DISTANCE - MORE TECHNICALLY

- The unicity distance of a cryptosystem is a number that indicates the amount of cryptotext required in order to uniquely determine the plaintext.
- It is a function of the length of the key used for encryption and the statistical nature of the plaintext language.
- Outcome: It can be shown that enough time it is guaranteed that any cipher can be broken given cryptotexts of such total length that unicity distance is 1.
- Formally, the unicity distance is given by

$$\frac{H(K)}{|M| - H(M)}$$

where

  - $H(k)$ is information content of the key.
  - $|M|$ je information content per symbol of the message assuming assuming that al symbols are equally likely.

UNICITY DISTANCE - MORE TECHNICALLY

- The unicity distance of a cryptosystem is a number that indicates the amount of cryptotext required in order to uniquely determine the plaintext.
- It is a function of the length of the key used for encryption and the statistical nature of the plaintext language.
- Outcome: It can be shown that enough time it is guaranteed that any cipher can be broken given cryptotexts of such total length that unicity distance is 1.
- Formally, the unicity distance is given by

$$\frac{H(K)}{|M| - H(M)}$$

where

  - $H(k)$ is information content of the key.
  - $|M|$ je information content per symbol of the message assuming assuming that al symbols are equally likely.
  - $H(M)$ is information content per symbol of the message.

EXAMPLE

- If K is chosen random, then $H(K) = |K|$, so for randomly chosen letters in English, $H(K) = |K| = \lg(26) = 4.7$ bits.

EXAMPLE

- If K is chosen random, then $H(K) = |K|$, so for randomly chosen letters in English, $H(K) = |K| = \lg(26) = 4.7$ bits.
- $H(M)$ has been empirically found to be 2.9 bits for English.

EXAMPLE

- If K is chosen random, then $H(K) = |K|$, so for randomly chosen letters in English, $H(K) = |K| = \lg(26) = 4.7$ bits.
- $H(M)$ has been empirically found to be 2.9 bits for English.
- Therefore the unicity distance for English is 1 when $|M| = (4.7/1.8)|K|$

ANAGRAMS – EXAMPLES

German:

IRI BRÄTER, GENF

ANAGRAMS – EXAMPLES

German:

IRI BRÄTER, GENF	Briefträgerin
FRANK PEKL, REGEN	...
PEER ASSSTIL, MELK	...
INGO DILMR, PEINE	...
EMIL REST, GERA	...
KARL SORDORT, PEINE	...

ANAGRAMS – EXAMPLES

German:

IRI BRÄTER, GENF	Briefträgerin
FRANK PEKL, REGEN	. . .
PEER ASSSTIL, MELK	. . .
INGO DILMR, PEINE	. . .
EMIL REST, GERA	. . .
KARL SORDORT, PEINE	. . .

English:

algorithms	logarithms
antagonist	stagnation
compressed	decompress
coordinate	decoration
creativity	reactivity
deductions	discounted
descriptor	predictors
impression	permission
introduces	reductions
procedures	reproduces

SOME SOLUTIONS

FRANK PEKL, REGEN

FRANK PEKL, REGEN Krankenpfleger
PEER ASTIL, MELK

FRANK PEKL, REGEN — Krankenpfleger

PEER ASTIL, MELK — Kapellmeister

INGO DILMR, PEINE

SOME SOLUTIONS

FRANK PEKL, REGEN	Krankenpfleger
PEER ASTIL, MELK	Kapellmeister
INGO DILMR, PEINE	Diplomengineer
EMIL REST, GERA	

FRANK PEKL, REGEN	Krankenpfleger
PEER ASTIL, MELK	Kapellmeister
INGO DILMR, PEINE	Diplomengineer
EMIL REST, GERA	Lagermeister
KARL SORDORT, PEINE	Personaldirector

APPENDIX I

FAMOUS CRYPTOGRAPHERS

FAMOUS CRYPTOGRAPHERS

- Girolamo Cardano (1501-1576) - father of probability theory

FAMOUS CRYPTOGRAPHERS

- Girolamo Cardano (1501-1576) - father of probability theory
- De la Bigotiere Viete (1540-1603) - father of modern algebra.

FAMOUS CRYPTOGRAPHERS

- Girolamo Cardano (1501-1576) - father of probability theory
- De la Bigotiere Viete (1540-1603) - father of modern algebra.
- Antoine Rosignol (father of Cryptology for France)

FAMOUS CRYPTOGRAPHERS

- Girolamo Cardano (1501-1576) - father of probability theory
- De la Bigotiere Viete (1540-1603) - father of modern algebra.
- Antoine Rosignol (father of Cryptology for France)
- John Wallis (1616-1703) (father of Cryptology for England)

FAMOUS CRYPTOGRAPHERS

- Girolamo Cardano (1501-1576) - father of probability theory
- De la Bigotiere Viete (1540-1603) - father of modern algebra.
- Antoine Rosignol (father of Cryptology for France)
- John Wallis (1616-1703) (father of Cryptology for England)
- Thomas Jefferson (1743-1826) - Father of American Cryptography)

FAMOUS CRYPTOGRAPHERS

- Girolamo Cardano (1501-1576) - father of probability theory
- De la Bigotiere Viete (1540-1603) - father of modern algebra.
- Antoine Rosignol (father of Cryptology for France)
- John Wallis (1616-1703) (father of Cryptology for England)
- Thomas Jefferson (1743-1826) - Father of American Cryptography)
- Charles Babbage (broke Vigenere cryptosystem - the inventor of the first universal computer).

FAMOUS CRYPTOGRAPHERS

- Girolamo Cardano (1501-1576) - father of probability theory
- De la Bigotiere Viete (1540-1603) - father of modern algebra.
- Antoine Rosignol (father of Cryptology for France)
- John Wallis (1616-1703) (father of Cryptology for England)
- Thomas Jefferson (1743-1826) - Father of American Cryptography)
- Charles Babbage (broke Vigenere cryptosystem - the inventor of the first universal computer).
- Allan Turing (broke ENIGMA, designed compute BOMBS, printed basic results on computer

FAMOUS CRYPTOGRAPHERS

- Girolamo Cardano (1501-1576) - father of probability theory
- De la Bigotiere Viete (1540-1603) - father of modern algebra.
- Antoine Rosignol (father of Cryptology for France)
- John Wallis (1616-1703) (father of Cryptology for England)
- Thomas Jefferson (1743-1826) - Father of American Cryptography)
- Charles Babbage (broke Vigenere cryptosystem - the inventor of the first universal computer).
- Allan Turing (broke ENIGMA, designed compute BOMBS, printed basic results on computer

CODEBOOKS CRYPTOGRAPHY

CODEBOOKS CRYPTOGRAPHY

- In the middle age, messages were mostly encrypted with "code books" (codebooks).

CODEBOOKS CRYPTOGRAPHY

- In the middle age, messages were mostly encrypted with "code books" (codebooks).
- In this set-up communicating parties, say Alice and Bob, shared some secret information, called the codebook.

CODEBOOKS CRYPTOGRAPHY

- In the middle age, messages were mostly encrypted with "code books" (codebooks).
- In this set-up communicating parties, say Alice and Bob, shared some secret information, called the codebook.
- Such a codebook can be a simple letter-to-letter substitution or a more complex word-by-word substitution.

CODEBOOKS CRYPTOGRAPHY

- In the middle age, messages were mostly encrypted with "code books" (codebooks).
- In this set-up communicating parties, say Alice and Bob, shared some secret information, called the codebook.
- Such a codebook can be a simple letter-to-letter substitution or a more complex word-by-word substitution.
- **Communication:** A sender encrypts her message using secret codebook and the receiver uses the same codebook to decrypt the encrypted message.

CODEBOOKS CRYPTOGRAPHY

- In the middle age, messages were mostly encrypted with "code books" (codebooks).
- In this set-up communicating parties, say Alice and Bob, shared some secret information, called the codebook.
- Such a codebook can be a simple letter-to-letter substitution or a more complex word-by-word substitution.
- **Communication:** A sender encrypts her message using secret codebook and the receiver uses the same codebook to decrypt the encrypted message.
- An eavesdropper cannot, in theory, decrypt the message because she does not posses the secret codebook.

CODEBOOKS CRYPTOGRAPHY

- In the middle age, messages were mostly encrypted with "code books" (codebooks).
- In this set-up communicating parties, say Alice and Bob, shared some secret information, called the codebook.
- Such a codebook can be a simple letter-to-letter substitution or a more complex word-by-word substitution.
- **Communication:** A sender encrypts her message using secret codebook and the receiver uses the same codebook to decrypt the encrypted message.
- An eavesdropper cannot, in theory, decrypt the message because she does not posses the secret codebook.
- A more modern term for "codebook" is the "key".

CODEBOOKS CRYPTOGRAPHY

- In the middle age, messages were mostly encrypted with "code books" (codebooks).
- In this set-up communicating parties, say Alice and Bob, shared some secret information, called the codebook.
- Such a codebook can be a simple letter-to-letter substitution or a more complex word-by-word substitution.
- **Communication:** A sender encrypts her message using secret codebook and the receiver uses the same codebook to decrypt the encrypted message.
- An eavesdropper cannot, in theory, decrypt the message because she does not posses the secret codebook.
- A more modern term for "codebook" is the "key".
- Codebooks were intensively used during the first World War. Some had up 100 000 encoding rules. The fact that allies were able to obtained huge codebooks from several destroyed war ships helped Allies much.

CODEBOOKS CRYPTOGRAPHY

- In the middle age, messages were mostly encrypted with "code books" (codebooks).
- In this set-up communicating parties, say Alice and Bob, shared some secret information, called the codebook.
- Such a codebook can be a simple letter-to-letter substitution or a more complex word-by-word substitution.
- **Communication:** A sender encrypts her message using secret codebook and the receiver uses the same codebook to decrypt the encrypted message.
- An eavesdropper cannot, in theory, decrypt the message because she does not posses the secret codebook.
- A more modern term for "codebook" is the "key".
- Codebooks were intensively used during the first World War. Some had up 100 000 encoding rules. The fact that allies were able to obtained huge codebooks from several destroyed war ships helped Allies much.
- Till recently it was assumed that secret codebooks are necessary for secret communication.

NOMENCLATORS

NOMENCLATORS

- For several centuries so called "Nomenclators" were used for encryption an decryption.

NOMENCLATORS

- For several centuries so called "Nomenclators" were used for encryption an decryption.
- Nomenclators were in use from the end of 14th century for 450 years.

NOMENCLATORS

- For several centuries so called "Nomenclators" were used for encryption an decryption.
- Nomenclators were in use from the end of 14th century for 450 years.
- Nomenclators combined a substitution cryptosystem (in which symbols were replaced by numbers) with codebook ciphers in which words were replaced by numbers.

NOMENCLATORS

- For several centuries so called "Nomenclators" were used for encryption an decryption.
- Nomenclators were in use from the end of 14th century for 450 years.
- Nomenclators combined a substitution cryptosystem (in which symbols were replaced by numbers) with codebook ciphers in which words were replaced by numbers.
- At the beginning codebooks had codes only for names of people (therefore such a name - nomenclators), later codes were used also for names of places and so on.

NOMENCLATORS

- For several centuries so called "Nomenclators" were used for encryption an decryption.
- Nomenclators were in use from the end of 14th century for 450 years.
- Nomenclators combined a substitution cryptosystem (in which symbols were replaced by numbers) with codebook ciphers in which words were replaced by numbers.
- At the beginning codebooks had codes only for names of people (therefore such a name - nomenclators), later codes were used also for names of places and so on.
- Some nomenclators had huge codebooks, up to 50 000 entries.

NOMENCLATORS

- For several centuries so called "Nomenclators" were used for encryption an decryption.
- Nomenclators were in use from the end of 14th century for 450 years.
- Nomenclators combined a substitution cryptosystem (in which symbols were replaced by numbers) with codebook ciphers in which words were replaced by numbers.
- At the beginning codebooks had codes only for names of people (therefore such a name - nomenclators), later codes were used also for names of places and so on.
- Some nomenclators had huge codebooks, up to 50 000 entries.
- Famous was the nomenclator designed by very famous French cryptologist Rosignol, for Ludvig XIV, that was not broken for several hundred of years.

NOMENCLATORS

- For several centuries so called "Nomenclators" were used for encryption an decryption.
- Nomenclators were in use from the end of 14th century for 450 years.
- Nomenclators combined a substitution cryptosystem (in which symbols were replaced by numbers) with codebook ciphers in which words were replaced by numbers.
- At the beginning codebooks had codes only for names of people (therefore such a name - nomenclators), later codes were used also for names of places and so on.
- Some nomenclators had huge codebooks, up to 50 000 entries.
- Famous was the nomenclator designed by very famous French cryptologist Rosignol, for Ludvig XIV, that was not broken for several hundred of years.
- It was the design of the telegraph and the need for *field ciphers* to be used in combat that ended the massive use of nomenclators and started a new history of cryptography dominated by polyalphabetic substitution cryptosystems.