

Cryptography

Michel Dubois

michel.dubois@esiea-ouest.fr

Last update: November 16, 2014



Table of contents

- 1 Introduction**
- 2 Mathematical background
- 3 History
- 4 In practice...



Table of contents

- 1 Introduction**
- 2 Mathematical background**
- 3 History**
- 4 In practice...**



Table of contents

- 1 Introduction**
- 2 Mathematical background**
- 3 History**
- 4 In practice...**



Table of contents

- 1 Introduction**
- 2 Mathematical background**
- 3 History**
- 4 In practice...**



Introduction

- 1** Introduction
- 2** Mathematical background
- 3** History
- 4** In practice...



What is cryptography?

1 Introduction

- What is cryptography?
- Why cryptography?
- Definitions



An enigma...

- ▶ Alice wishes to send a gift to Bob.
- ▶ She does not trust the conveyor.
- ▶ how should she do to be sure that bob receives his gift?



An enigma...

- ▶ Alice wishes to send a gift to Bob.
- ▶ She does not trust the conveyor.
- ▶ how should she do to be sure that bob receives his gift?

without change
without loss



An enigma...

- ▶ Alice wishes to send a gift to Bob.
- ▶ She does not trust the conveyor.
- ▶ how should she do to be sure that bob receives his gift?
 - ▶ without change
 - ▶ without open



An enigma...

- ▶ Alice wishes to send a gift to Bob.
- ▶ She does not trust the conveyor.
- ▶ how should she do to be sure that bob receives his gift?
 - ▶ without change
 - ▶ without open



An enigma...

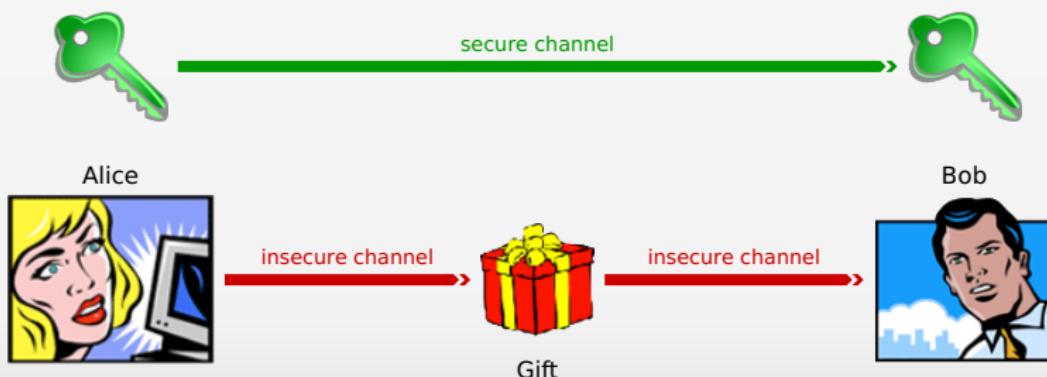
- ▶ Alice wishes to send a gift to Bob.
- ▶ She does not trust the conveyor.
- ▶ how should she do to be sure that bob receives his gift?
 - ▶ without change
 - ▶ without open



An enigma...

Symmetric solution: one secret shared key

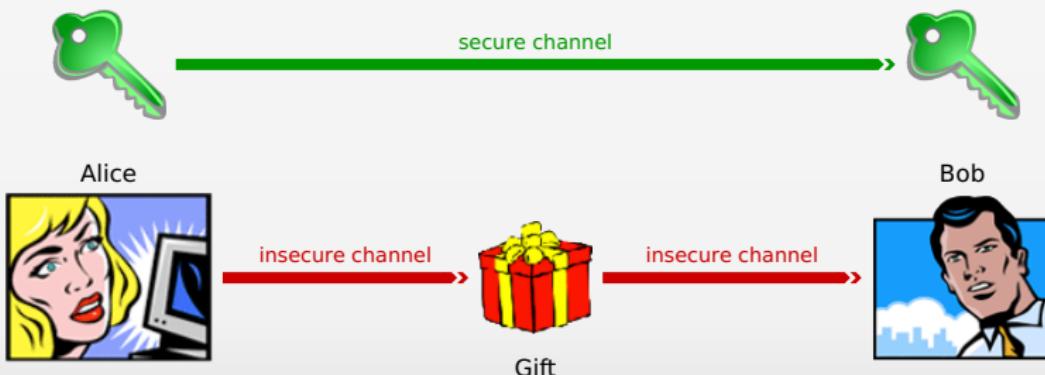
- ▶ Alice chooses a padlock and sends a copy of the key to Bob.
- ▶ Alice closes the gift package with the padlock and sends it to Bob.
- ▶ Bob receives the packet, opens it with his key and gets his gift.



An enigma...

Symmetric solution: one secret shared key

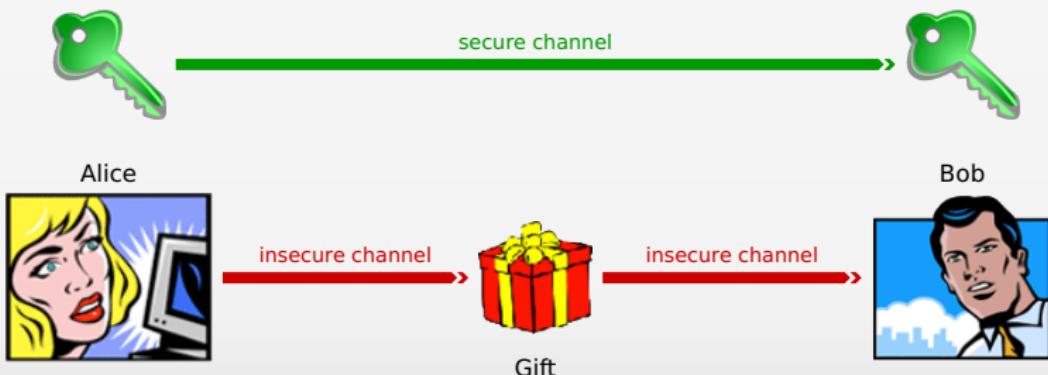
- ▶ Alice chooses a padlock and sends a copy of the key to Bob.
- ▶ Alice closes the gift package with the padlock and sends it to Bob.
- ▶ Bob receives the packet, opens it with his key and gets his gift.



An enigma...

Symmetric solution: one secret shared key

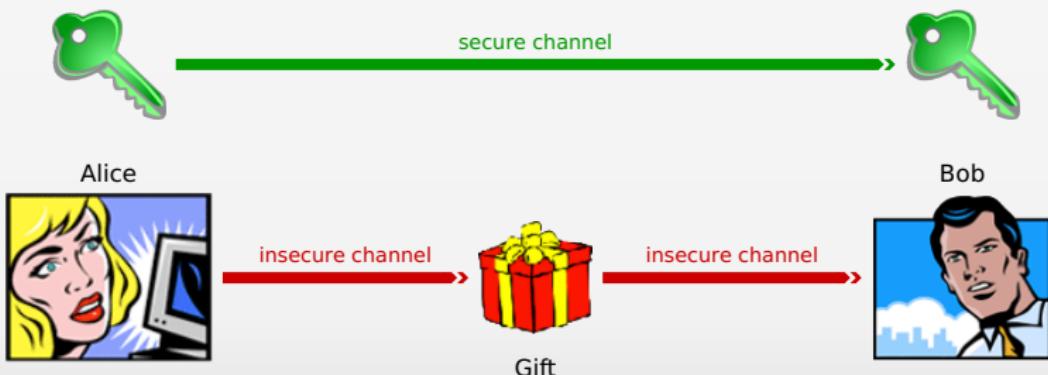
- ▶ Alice chooses a padlock and sends a copy of the key to Bob.
- ▶ Alice closes the gift package with the padlock and sends it to Bob.
- ▶ Bob receives the packet, opens it with his key and gets his gift.



An enigma...

Symmetric solution: one secret shared key

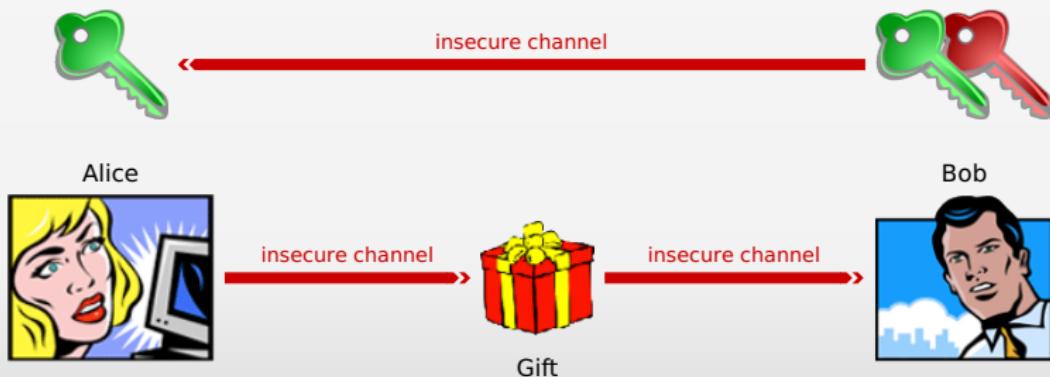
- ▶ Alice chooses a padlock and sends a copy of the key to Bob.
- ▶ Alice closes the gift package with the padlock and sends it to Bob.
- ▶ **Bob receives the packet, opens it with his key and gets his gift.**



An enigma...

Asymmetric solution: a public key to encrypt and a private key to decrypt

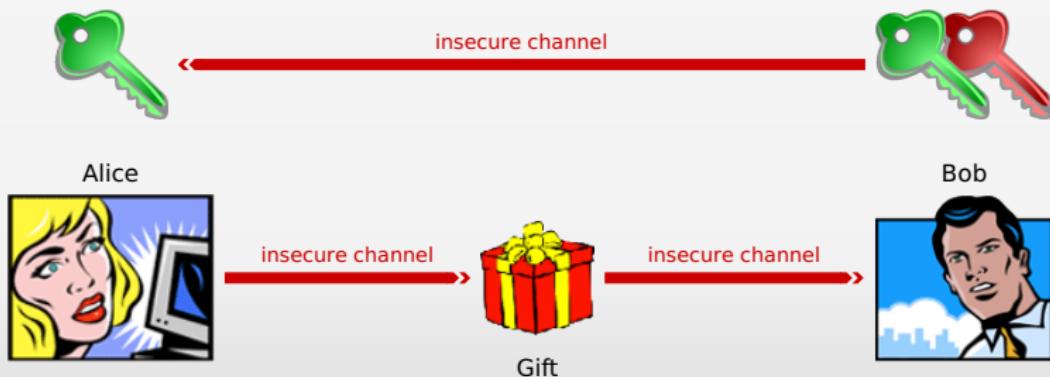
- ▶ Bob choose a padlock.
- ▶ He sends the padlock open to Alice and keeps the key.
- ▶ When Alice wants to send her gift to Bob, she closes the gift package with the padlock and sends it to Bob.
- ▶ Bob receives the packet, opens it with his key and gets the gift.



An enigma...

Asymmetric solution: a public key to encrypt and a private key to decrypt

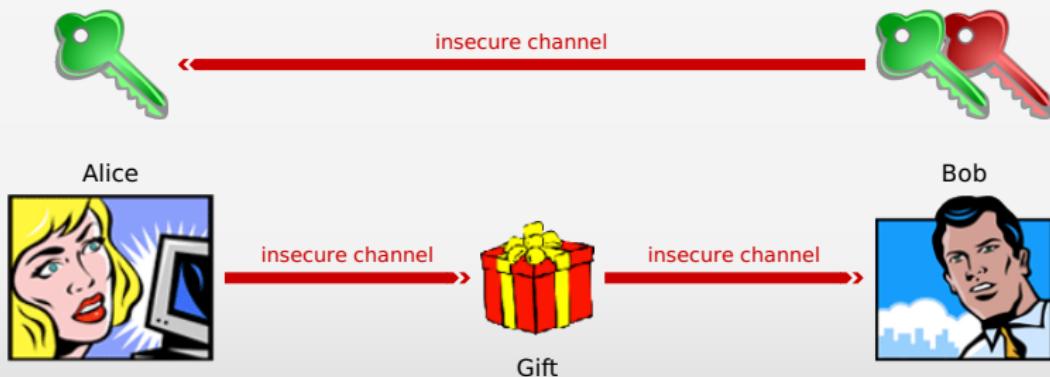
- ▶ Bob choose a padlock.
- ▶ He sends the padlock open to Alice and keeps the key.
- ▶ When Alice wants to send her gift to Bob, she closes the gift package with the padlock and sends it to Bob.
- ▶ Bob receives the packet, opens it with his key and gets the gift.



An enigma...

Asymmetric solution: a public key to encrypt and a private key to decrypt

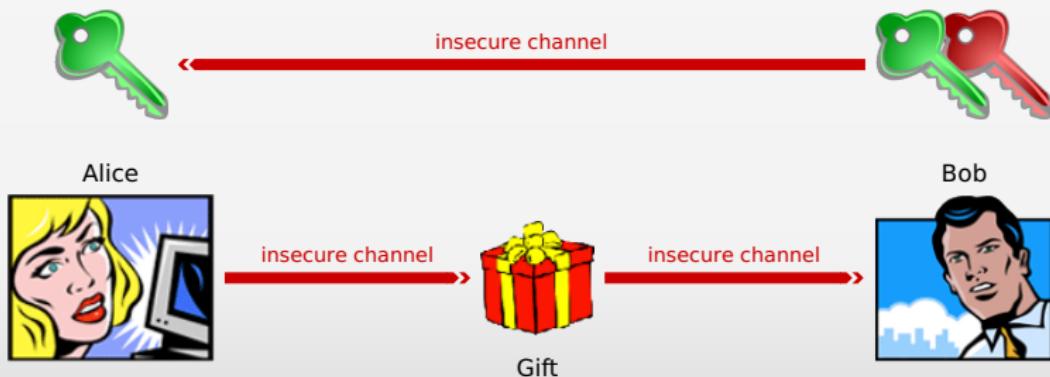
- ▶ Bob choose a padlock.
- ▶ He sends the padlock open to Alice and keeps the key.
- ▶ When Alice wants to send her gift to Bob, she closes the gift package with the padlock and sends it to Bob.
- ▶ Bob receives the packet, opens it with his key and gets the gift.



An enigma...

Asymmetric solution: a public key to encrypt and a private key to decrypt

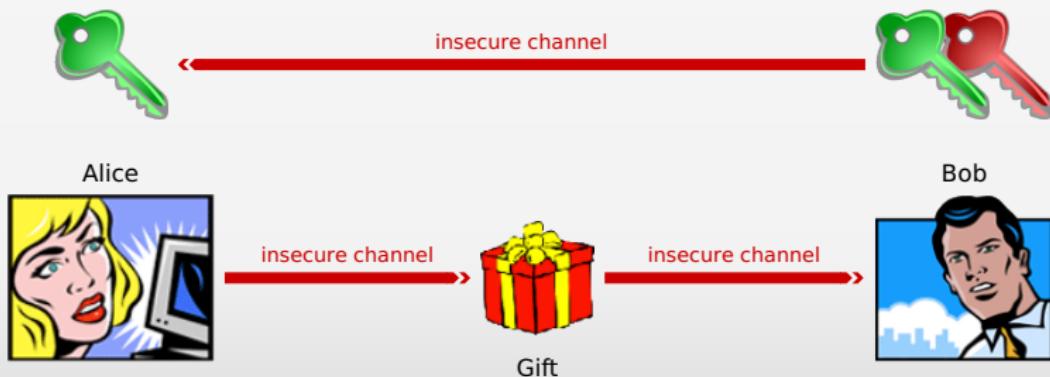
- ▶ Bob choose a padlock.
- ▶ He sends the padlock open to Alice and keeps the key.
- ▶ When Alice wants to send her gift to Bob, she closes the gift package with the padlock and sends it to Bob.
- ▶ Bob receives the packet, opens it with his key and gets the gift.



An enigma...

Asymmetric solution: a public key to encrypt and a private key to decrypt

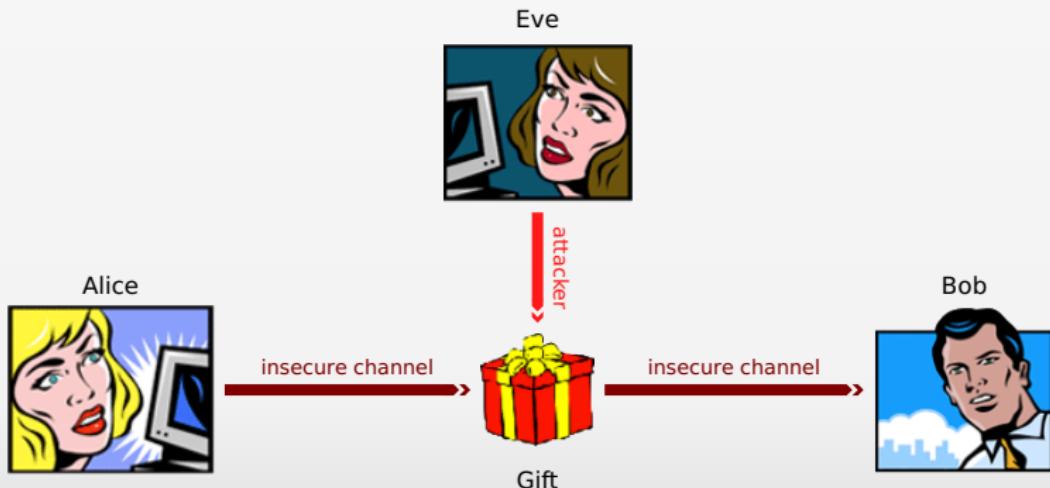
- ▶ Bob choose a padlock.
- ▶ He sends the padlock open to Alice and keeps the key.
- ▶ When Alice wants to send her gift to Bob, she closes the gift package with the padlock and sends it to Bob.
- ▶ Bob receives the packet, opens it with his key and gets the gift.



Fundamentals

Basis

The fundamental purpose of cryptography is to enable two people to **communicate** through an insecure channel so that a stranger can not intercept the communication.



Why cryptography?

1 Introduction

- What is cryptography?
- Why cryptography?
- Definitions



Cryptography is used for...

Ensure **confidentiality**

Confidentiality

Confidentiality is the property that information is neither **available** nor **disclosed** to persons, entities or processes are not allowed.

► Confidentiality

Keeping the information hidden for all those who are not allowed to access it.

► Anonymity

Concealing the identity of an entity involved in a process.

Cryptography is used for...

Ensure **confidentiality**

Confidentiality

Confidentiality is the property that information is neither **available** nor **disclosed** to persons, entities or processes are not allowed.

► Confidentiality



Keeping the information hidden for all those who are not allowed to access it.

► Anonymity

Concealing the identity of an entity involved in a process.

Cryptography is used for...

Ensure **confidentiality**

Confidentiality

Confidentiality is the property that information is neither **available** nor **disclosed** to persons, entities or processes are not allowed.

► Confidentiality



Keeping the information hidden for all those who are not allowed to access it.

► Anonymity

Concealing the identity of an entity involved in a process.

Cryptography is used for...

Ensure **confidentiality**

Confidentiality

Confidentiality is the property that information is neither **available** nor **disclosed** to persons, entities or processes are not allowed.

► Confidentiality



► Anonymity



Keeping the information hidden for all those who are not allowed to access it.

Concealing the identity of an entity involved in a process.

Cryptography is used for...

Ensure **confidentiality**

Confidentiality

Confidentiality is the property that information is neither **available** nor **disclosed** to persons, entities or processes are not allowed.

► Confidentiality



► Anonymity



Keeping the information hidden for all those who are not allowed to access it.

Concealing the identity of an entity involved in a process.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

► Data integrity

Ensuring that the information has not been altered by unauthorized or unknown entity.

► Validation

A means to provide the authorization to use or manipulate information or resources.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

► Data integrity



► Validation

A means to provide the authorization to use or manipulate information or resources.

Ensuring that the information has not been altered by unauthorized or unknown entity.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

► Data integrity



► Validation

A means to provide the authorization to use or manipulate information or resources.

Ensuring that the information has not been altered by unauthorized or unknown entity.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

► Data integrity



► Validation



Ensuring that the information has not been altered by unauthorized or unknown entity.

A means to provide the authorization to use or manipulate information or resources.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

► Data integrity



► Validation



Ensuring that the information has not been altered by unauthorized or unknown entity.

A means to provide the authorization to use or manipulate information or resources.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

- ▶ Witnessing



- ▶ Authorization

Transfer to another entity of approbation to use or manipulate information or resources.

Verifying the creation or existence of information by an entity other than the creator.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

► Witnessing



► Authorization

Transfer to another entity of approbation to use or manipulate information or resources.

Verifying the creation or existence of information by an entity other than the creator.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

► Witnessing



Verifying the creation or existence of information by an entity other than the creator.

► Authorization



Transfer to another entity of approbation to use or manipulate information or resources.

Cryptography is used for...

Ensure the **integrity**

Integrity

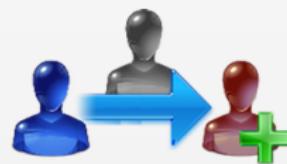
Integrity is the prevention of an **unauthorized modification** of the information.

► Witnessing



Verifying the creation or existence of information by an entity other than the creator.

► Authorization



Transfer to another entity of approbation to use or manipulate information or resources.

Cryptography is used for...

Ensure the **integrity**

Integrity

Integrity is the prevention of an **unauthorized modification** of the information.

► Witnessing



Verifying the creation or existence of information by an entity other than the creator.

► Authorization



Transfer to another entity of approbation to use or manipulate information or resources.

Cryptography is used for...

Ensure the **authentication**

Authentication

Proving its identity by what we **know**, by what **one is**, by what we **have**, by what we **do or where one is**.

► Entity authentication

Corroboration of the identity of an entity.

► Message authentication

Corroboration of the source of an information.

► Timestamping

Recording the time of creation or existence of information.

Cryptography is used for...

Ensure the **authentication**

Authentication

Proving its identity by what we **know**, by what **one is**, by what we **have**, by what we **do or where one is**.

► Entity authentication



Corroboration of the identity of an entity.

► Message authentication

Corroboration of the source of an information.

► Timestamping

Recording the time of creation or existence of information.

Cryptography is used for...

Ensure the **authentication**

Authentication

Proving its identity by what we **know**, by what **one is**, by what we **have**, by what we **do or where one is**.

► Entity authentication



Corroboration of the identity of an entity.

► Message authentication

Corroboration of the source of an information.

► Timestamping

Recording the time of creation or existence of information.

Cryptography is used for...

Ensure the **authentication**

Authentication

Proving its identity by what we **know**, by what **one is**, by what we **have**, by what we **do or where one is**.

► Entity authentication



Corroboration of the identity of an entity.

► Message authentication



Corroboration of the source of an information.

► Timestamping

Recording the time of creation or existence of information.

Cryptography is used for...

Ensure the **authentication**

Authentication

Proving its identity by what we **know**, by what **one is**, by what we **have**, by what we **do or where one is**.

- ▶ Entity authentication



Corroboration of the identity of an entity.

- ▶ Message authentication



Corroboration of the source of an information.

- ▶ Timestamping

Recording the time of creation or existence of information.

Cryptography is used for...

Ensure the **authentication**

Authentication

Proving its identity by what we **know**, by what **one is**, by what we **have**, by what we **do or where one is**.

▶ Entity authentication



Corroboration of the identity of an entity.

▶ Message authentication



Corroboration of the source of an information.

▶ Timestamping



Recording the time of creation or existence of information.

Cryptography is used for...

Ensure the **authentication**

Authentication

Proving its identity by what we **know**, by what **one is**, by what we **have**, by what we **do or where one is**.

▶ Entity authentication



Corroboration of the identity of an entity.

▶ Message authentication



Corroboration of the source of an information.

▶ Timestamping



Recording the time of creation or existence of information.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Non-repudiation

► Signature

Preventing the denial of previous commitments or actions.

A means to bind information to an entity.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Non-repudiation



► Signature

A means to bind information to an entity.

Preventing the denial of previous commitments or actions.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

- ▶ Non-repudiation



- ▶ Signature

A means to bind information to an entity.

Preventing the denial of previous commitments or actions.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Non-repudiation



► Signature



Preventing the denial of previous commitments or actions.

A means to bind information to an entity.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Non-repudiation



Preventing the denial of previous commitments or actions.

► Signature



A means to bind information to an entity.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Receipt



► Confirmation

Acknowledgment that services have been provided.

Acknowledgment that information transmitted has been received.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Receipt



► Confirmation

Acknowledgment that services have been provided.

Acknowledgment that information transmitted has been received.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Receipt



Acknowledgment that information transmitted has been received.

► Confirmation



Acknowledgment that services have been provided.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Receipt



Acknowledgment that information transmitted has been received.

► Confirmation



Acknowledgment that services have been provided.

Cryptography is used for...

Ensure computer **transactions**

Computer transaction

An computer transaction consist to execute a **coherent operation** composed of several **unitary tasks**. The operation is valid only if all the unitary tasks are performed correctly.

► Receipt



Acknowledgment that information transmitted has been received.

► Confirmation



Acknowledgment that services have been provided.

Cryptography is used for...

Ensure the **access control**

Access control

Access control consists in **checking** whether an entity requesting access to a resource **has the rights** to do so. Access control provides the ability to access **physical** and **logical** resources.

► Access control

Restricting access to resources to privileged entities.

► Certification

Endorsement of information by a trusted entity.

► Revocation

Retraction of certification or authorization.

Cryptography is used for...

Ensure the **access control**

Access control

Access control consists in **checking** whether an entity requesting access to a resource **has the rights** to do so. Access control provides the ability to access **physical** and **logical** resources.

► Access control



Restricting access to resources to privileged entities.

► Certification

Endorsement of information by a trusted entity.

► Revocation

Retraction of certification or authorization.

Cryptography is used for...

Ensure the **access control**

Access control

Access control consists in **checking** whether an entity requesting access to a resource **has the rights** to do so. Access control provides the ability to access **physical** and **logical** resources.

► Access control



► Certification

► Revocation

Endorsement of information by a trusted entity.

Retraction of certification or authorization.

Restricting access to resources to privileged entities.

Cryptography is used for...

Ensure the **access control**

Access control

Access control consists in **checking** whether an entity requesting access to a resource **has the rights** to do so. Access control provides the ability to access **physical** and **logical** resources.

► Access control



Restricting access to resources to privileged entities.

► Certification



Endorsement of information by a trusted entity.

► Revocation

Retraction of certification or authorization.

Cryptography is used for...

Ensure the **access control**

Access control

Access control consists in **checking** whether an entity requesting access to a resource **has the rights** to do so. Access control provides the ability to access **physical** and **logical** resources.

► Access control



Restricting access to resources to privileged entities.

► Certification



Endorsement of information by a trusted entity.

► Revocation

Retraction of certification or authorization.

Cryptography is used for...

Ensure the **access control**

Access control

Access control consists in **checking** whether an entity requesting access to a resource **has the rights** to do so. Access control provides the ability to access **physical** and **logical** resources.

► Access control



Restricting access to resources to privileged entities.

► Certification



Endorsement of information by a trusted entity.

► Revocation



Retraction of certification or authorization.

Cryptography is used for...

Ensure the **access control**

Access control

Access control consists in **checking** whether an entity requesting access to a resource **has the rights** to do so. Access control provides the ability to access **physical** and **logical** resources.

► Access control



► Certification



► Revocation



Restricting access to resources to privileged entities.

Endorsement of information by a trusted entity.

Retraction of certification or authorization.

Cryptography is used for...

Ensure the **ownership**

► Ownership



A means to provide an entity with the legal right to use or transfer a resource to others.

Cryptography is used for...

Ensure the **ownership**

- ▶ Ownership



A means to provide an entity with the legal right to use or transfer a resource to others.

Definitions

1 Introduction

- What is cryptography?
- Why cryptography?
- Definitions



Information and Cryptography

Information

In the world of cryptography the concept of information defines an understandable quantity.

Cryptography

- ▶ *κρυπτός*: hidden, secret or dark
- ▶ *γράφειν*: act of drawing

Set of mathematical methods implemented to ensure the protection of a message.

Information and Cryptography

Information

In the world of cryptography the concept of information defines an understandable quantity.

Cryptography

- ▶ **κρυπτός**: hidden, secret or dark
- ▶ **γράφειν**: act of drawing

Set of mathematical methods implemented to ensure the protection of a message.

Ciphering and Deciphering

Ciphering

Ciphering consists, using an encryption key, to transform a clear text to a cipher text.

Deciphering

Deciphering is the reciprocal operation of the ciphering. It consists, **knowing** the encryption key, to transform a cipher text to the corresponding clear text.

Cryptogram

This is the encrypted message.

Ciphering and Deciphering

Ciphering

Ciphering consists, using an encryption key, to transform a clear text to a cipher text.

Deciphering

Deciphering is the reciprocal operation of the ciphering. It consists, **knowing** the encryption key, to transform a cipher text to the corresponding clear text.

Cryptogram

This is the encrypted message.

Ciphering and Deciphering

Ciphering

Ciphering consists, using an encryption key, to transform a clear text to a cipher text.

Deciphering

Deciphering is the reciprocal operation of the ciphering. It consists, **knowing** the encryption key, to transform a cipher text to the corresponding clear text.

Cryptogram

This is the encrypted message.

Decryption and Cryptanalysis

Decryption

Decryption consist to recover the clear text from a cipher text without knowing the encryption key.

Cryptanalysis

Cryptanalysis is the study of mathematical tools to perform the decryption of a cryptogram.

The Cyrillic Projector is a sculpture created by American artist James Sanborn in the early 1990s, and was purchased by the University of North Carolina at Charlotte in 1997. It is currently installed between the campus Friday and Fretwell Buildings.



Decryption and Cryptanalysis

Decryption

Decryption consist to recover the clear text from a cipher text without knowing the encryption key.

Cryptanalysis

Cryptanalysis is the study of mathematical tools to perform the decryption of a cryptogram.

The Cyrillic Projector is a sculpture created by American artist James Sanborn in the early 1990s, and was purchased by the University of North Carolina at Charlotte in 1997. It is currently installed between the campus Friday and Fretwell Buildings.



Cryptology

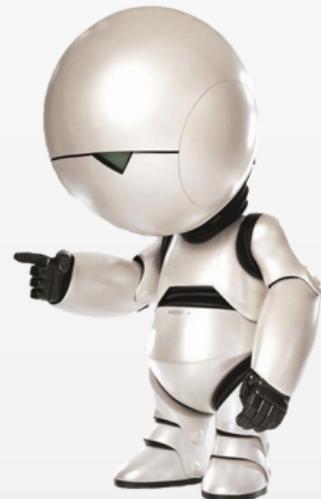
Cryptology = Cryptography + Cryptanalysis

Kryptos is a sculpture by American artist Jim Sanborn located on the grounds of the Central Intelligence Agency (CIA) in Langley, Virginia. Since its dedication on November 3, 1990, there has been much speculation about the meaning of the encrypted messages it bears. Of the four sections, three have been solved, with the fourth remaining one of the most famous unsolved codes in the world. The sculpture continues to provide a diversion for some employees of the CIA and other cryptanalysts attempting to decrypt the messages.



Mathematical background

- 1** Introduction
- 2** Mathematical background
- 3** History
- 4** In practice...



Notation

2 Mathematical background

- Notation
- Mathematics for cryptography
 - Analysis
 - Abstract algebra
 - Number theory
 - Boolean function
- Information Theory
- Complexity theory



Notation

- ▶ \mathbb{A} defines the finite set called **alphabet definition**
- ▶ \mathbb{P} defines the finite set called **plain text space**
- ▶ \mathbb{C} defines the finite set called **cipher text space**
- ▶ \mathbb{K} defines the finite set called **key space**
- ▶ \mathbb{E} and \mathbb{D} define respectively the finite set of encryption functions and the finite set of decryption functions.
- ▶ If $\forall e \in \mathbb{K}$ there is a function $E_e \in \mathbb{E} : \mathbb{P} \mapsto \mathbb{C}$ then E_e is called **ciphering function**
- ▶ If $\forall d \in \mathbb{K}$ there is a function $D_d \in \mathbb{D} : \mathbb{C} \mapsto \mathbb{P}$ then D_d is called **deciphering function**

A cryptosystem is the sextuplet S such that $S = \{\mathbb{A}, \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{E}, \mathbb{D}\}$

Notation

- ▶ \mathbb{A} defines the finite set called **alphabet definition**
- ▶ \mathbb{P} defines the finite set called **plain text space**
- ▶ \mathbb{C} defines the finite set called **cipher text space**
- ▶ \mathbb{K} defines the finite set called **key space**
- ▶ \mathbb{E} and \mathbb{D} define respectively the finite set of **encryption** functions and the finite set of **decryption** functions.
- ▶ If $\forall e \in \mathbb{K}$ there is a function $E_e \in \mathbb{E} : \mathbb{P} \mapsto \mathbb{C}$ then E_e is called **ciphering function**
- ▶ If $\forall d \in \mathbb{K}$ there is a function $D_d \in \mathbb{D} : \mathbb{C} \mapsto \mathbb{P}$ then D_d is called **deciphering function**

A cryptosystem is the sextuplet S such that $S = \{\mathbb{A}, \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{E}, \mathbb{D}\}$

Notation

- ▶ \mathbb{A} defines the finite set called **alphabet definition**
- ▶ \mathbb{P} defines the finite set called **plain text space**
- ▶ \mathbb{C} defines the finite set called **cipher text space**
- ▶ \mathbb{K} defines the finite set called **key space**
- ▶ \mathbb{E} and \mathbb{D} define respectively the finite set of **encryption** functions and the finite set of **decryption** functions.
- ▶ if $\forall e \in \mathbb{K}$ there is a function $E_e \in \mathbb{E} : \mathbb{P} \mapsto \mathbb{C}$ then E_e is called **ciphering function**
- ▶ if $\forall d \in \mathbb{K}$ there is a function $D_d \in \mathbb{D} : \mathbb{C} \mapsto \mathbb{P}$ then D_d is called **deciphering function**

A cryptosystem is the sextuplet S such that $S = \{\mathbb{A}, \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{E}, \mathbb{D}\}$

Notation

- ▶ \mathbb{A} defines the finite set called **alphabet definition**
- ▶ \mathbb{P} defines the finite set called **plain text space**
- ▶ \mathbb{C} defines the finite set called **cipher text space**
- ▶ \mathbb{K} defines the finite set called **key space**
- ▶ \mathbb{E} and \mathbb{D} define respectively the finite set of **encryption** functions and the finite set of **decryption** functions.
- ▶ if $\forall e \in \mathbb{K}$ there is a function $E_e \in \mathbb{E} : \mathbb{P} \mapsto \mathbb{C}$ then E_e is called **ciphering function**
- ▶ if $\forall d \in \mathbb{K}$ there is a function $D_d \in \mathbb{D} : \mathbb{C} \mapsto \mathbb{P}$ then D_d is called **deciphering function**

A cryptosystem is the sextuplet S such that $S = \{\mathbb{A}, \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{E}, \mathbb{D}\}$

Notation

- ▶ \mathbb{A} defines the finite set called **alphabet definition**
- ▶ \mathbb{P} defines the finite set called **plain text space**
- ▶ \mathbb{C} defines the finite set called **cipher text space**
- ▶ \mathbb{K} defines the finite set called **key space**
- ▶ \mathbb{E} and \mathbb{D} define respectively the finite set of **encryption** functions and the finite set of **decryption** functions.
- ▶ if $\forall e \in \mathbb{K}$ there is a function $E_e \in \mathbb{E} : \mathbb{P} \mapsto \mathbb{C}$ then E_e is called **ciphering function**
- ▶ if $\forall d \in \mathbb{K}$ there is a function $D_d \in \mathbb{D} : \mathbb{C} \mapsto \mathbb{P}$ then D_d is called **deciphering function**

A cryptosystem is the sextuplet \mathbb{S} such that $\mathbb{S} = \{\mathbb{A}, \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{E}, \mathbb{D}\}$

Notation

- ▶ \mathbb{A} defines the finite set called **alphabet definition**
- ▶ \mathbb{P} defines the finite set called **plain text space**
- ▶ \mathbb{C} defines the finite set called **cipher text space**
- ▶ \mathbb{K} defines the finite set called **key space**
- ▶ \mathbb{E} and \mathbb{D} define respectively the finite set of **encryption** functions and the finite set of **decryption** functions.
- ▶ if $\forall e \in \mathbb{K}$ there is a function $E_e \in \mathbb{E} : \mathbb{P} \mapsto \mathbb{C}$ then E_e is called **ciphering function**
- ▶ if $\forall d \in \mathbb{K}$ there is a function $D_d \in \mathbb{D} : \mathbb{C} \mapsto \mathbb{P}$ then D_d is called **deciphering function**

A cryptosystem is the sextuplet \mathbb{S} such that $\mathbb{S} = \{\mathbb{A}, \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{E}, \mathbb{D}\}$

Notation

- ▶ \mathbb{A} defines the finite set called **alphabet definition**
- ▶ \mathbb{P} defines the finite set called **plain text space**
- ▶ \mathbb{C} defines the finite set called **cipher text space**
- ▶ \mathbb{K} defines the finite set called **key space**
- ▶ \mathbb{E} and \mathbb{D} define respectively the finite set of **encryption** functions and the finite set of **decryption** functions.
- ▶ if $\forall e \in \mathbb{K}$ there is a function $E_e \in \mathbb{E} : \mathbb{P} \mapsto \mathbb{C}$ then E_e is called **ciphering function**
- ▶ if $\forall d \in \mathbb{K}$ there is a function $D_d \in \mathbb{D} : \mathbb{C} \mapsto \mathbb{P}$ then D_d is called **deciphering function**

A cryptosystem is the sextuplet \mathbb{S} such that $\mathbb{S} = \{\mathbb{A}, \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{E}, \mathbb{D}\}$

Notation

- ▶ \mathbb{A} defines the finite set called **alphabet definition**
- ▶ \mathbb{P} defines the finite set called **plain text space**
- ▶ \mathbb{C} defines the finite set called **cipher text space**
- ▶ \mathbb{K} defines the finite set called **key space**
- ▶ \mathbb{E} and \mathbb{D} define respectively the finite set of **encryption** functions and the finite set of **decryption** functions.
- ▶ if $\forall e \in \mathbb{K}$ there is a function $E_e \in \mathbb{E} : \mathbb{P} \mapsto \mathbb{C}$ then E_e is called **ciphering function**
- ▶ if $\forall d \in \mathbb{K}$ there is a function $D_d \in \mathbb{D} : \mathbb{C} \mapsto \mathbb{P}$ then D_d is called **deciphering function**

A cryptosystem is the sextuplet \mathbb{S} such that $\mathbb{S} = \{\mathbb{A}, \mathbb{P}, \mathbb{C}, \mathbb{K}, \mathbb{E}, \mathbb{D}\}$

Cryptographic scheme

- ▶ A **cryptographic scheme** is a set $\{E_e : e \in \mathbb{K}\}$ of ciphering functions and a corresponding set $\{D_d : d \in \mathbb{K}\}$ of deciphering functions with the property that $\forall e \in \mathbb{K}$ there is a unique key $d \in \mathbb{K}$ such that $D_d = E_e^{-1}$. That is $D_d(E_e(p)) = p, \quad \forall p \in \mathbb{P}$.
- ▶ In this case (e, d) means the key pair.
- ▶ A cryptographic scheme is called **breakable** if a third party, without prior knowledge of the key pair (e, d) , can systematically recover plaintext from the ciphertext in a reasonable time.

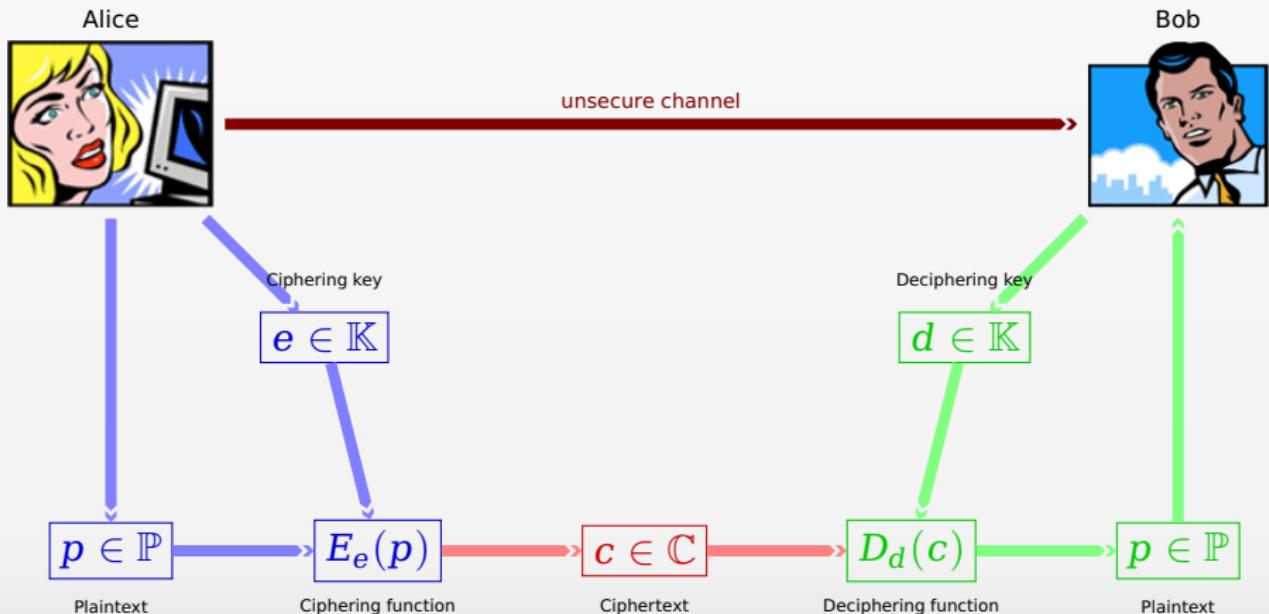
Cryptographic scheme

- ▶ A **cryptographic scheme** is a set $\{E_e : e \in \mathbb{K}\}$ of ciphering functions and a corresponding set $\{D_d : d \in \mathbb{K}\}$ of deciphering functions with the property that $\forall e \in \mathbb{K}$ there is a unique key $d \in \mathbb{K}$ such that $D_d = E_e^{-1}$. That is $D_d(E_e(p)) = p, \quad \forall p \in \mathbb{P}$.
- ▶ In this case (e, d) means the key pair.
- ▶ A cryptographic scheme is called **breakable** if a third party, without prior knowledge of the key pair (e, d) , can systematically recover plaintext from the ciphertext in a reasonable time.

Cryptographic scheme

- ▶ A **cryptographic scheme** is a set $\{E_e : e \in \mathbb{K}\}$ of ciphering functions and a corresponding set $\{D_d : d \in \mathbb{K}\}$ of deciphering functions with the property that $\forall e \in \mathbb{K}$ there is a unique key $d \in \mathbb{K}$ such that $D_d = E_e^{-1}$. That is $D_d(E_e(p)) = p, \quad \forall p \in \mathbb{P}$.
- ▶ In this case (e, d) means the key pair.
- ▶ A cryptographic scheme is called **breakable** if a third party, without prior knowledge of the key pair (e, d) , can systematically recover plaintext from the ciphertext in a reasonable time.

Ciphering process



Mathematics for cryptography

2 Mathematical background

- Notation
- Mathematics for cryptography
 - Analysis
 - Abstract algebra
 - Number theory
 - Boolean function
- Information Theory
- Complexity theory



Sage

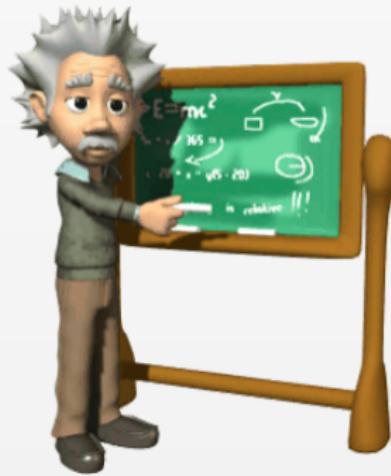
Tutorial: Sage

Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages into a common Python-based interface. <http://www.sagemath.org/>

- ① Download and install Sage on your desktop
- ② Study the guide tour at <http://www.sagemath.org/doc/tutorial/tour.html>
- ③ Study the group theory tutorial at
http://www.sagemath.org/doc/thematic_tutorials/group_theory.html



Analysis

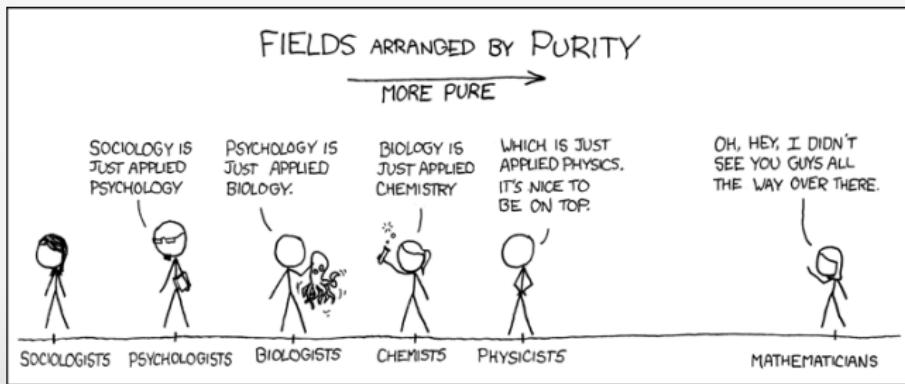


Analysis

Definition

Analysis is a branch of pure mathematics that includes the theories of differentiation, integration and measure, limits, infinite series, and analytic functions.

Analysis may be conventionally distinguished from geometry. However, theories of analysis can be applied to any space of mathematical objects that has a definition of nearness or distance.



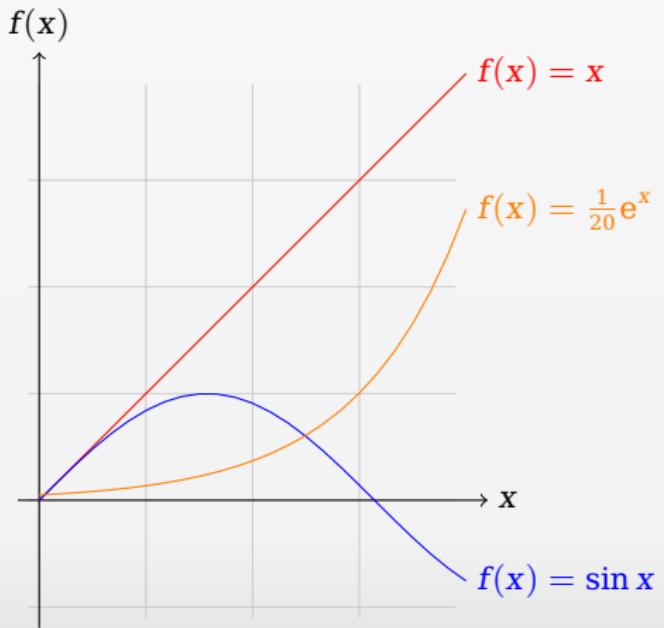
Analysis

Functions

Definition

A function is defined by two sets \mathbb{X} and \mathbb{Y} and a rule f which assigns to each element $x \in \mathbb{X}$ precisely one element $y \in \mathbb{Y}$ such that:

$$f : \begin{cases} \mathbb{X} \mapsto \mathbb{Y} \\ f(x) = y \end{cases}$$



Analysis

Functions

Properties

- ▶ The set of all inputs to a particular function is called its **domain**
- ▶ The set of all outputs of a particular function is called its **image**
- ▶ The **codomain** is the set into which all of the output of the function is constrained to fall
- ▶ For every function its codomain contains its image

Example

For a function $f : \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = x^2$ the *codomain* of f is \mathbb{R} , but f does not map to any negative number. Thus the *image* of f , denoted $\text{Im}(f)$, is the set \mathbb{R}_+ that is the interval $[0, \infty]$.

Analysis

Functions

Properties

- ▶ The set of all inputs to a particular function is called its **domain**
- ▶ The set of all outputs of a particular function is called its **image**
- ▶ The **codomain** is the set into which all of the output of the function is constrained to fall
- ▶ For every function its codomain includes its image.

Example

For a function $f: \mathbb{R} \rightarrow \mathbb{R}$ defined by $f(x) = x^2$ the *codomain* of f is \mathbb{R} , but f does not map to any negative number. Thus the *image* of f , denoted $\text{Im}(f)$, is the set \mathbb{R}_+ that is the interval $[0, \infty]$.

Analysis

Functions

Properties

- ▶ The set of all inputs to a particular function is called its **domain**
- ▶ The set of all outputs of a particular function is called its **image**
- ▶ The **codomain** is the set into which all of the output of the function is constrained to fall
- ▶ For every function its codomain **includes** its image.

Example

For a function $f : \mathbb{R} \mapsto \mathbb{R}$ defined by $f(x) = x^2$ the **codomain** of f is \mathbb{R} , but f does not map to any negative number. Thus the **image** of f , denoted $Im(f)$, is the set \mathbb{R}_0^+ that is the interval $[0, \infty]$

Analysis

Functions

Properties

- ▶ The set of all inputs to a particular function is called its **domain**
- ▶ The set of all outputs of a particular function is called its **image**
- ▶ The **codomain** is the set into which all of the output of the function is constrained to fall
- ▶ For every function its codomain **includes** its image.

Example

For a function $f : \mathbb{R} \mapsto \mathbb{R}$ defined by $f(x) = x^2$ the *codomain* of f is \mathbb{R} , but f does not map to any negative number. Thus the *image* of f , denoted $Im(f)$, is the set \mathbb{R}_0^+ that is the interval $[0, \infty[$

Analysis

Functions

Properties

- ▶ The set of all inputs to a particular function is called its **domain**
- ▶ The set of all outputs of a particular function is called its **image**
- ▶ The **codomain** is the set into which all of the output of the function is constrained to fall
- ▶ For every function its codomain **includes** its image.

Example

For a function $f : \mathbb{R} \mapsto \mathbb{R}$ defined by $f(x) = x^2$ the *codomain* of f is \mathbb{R} , but f does not map to any negative number. Thus the *image* of f , denoted $Im(f)$, is the set \mathbb{R}_0^+ that is the interval $[0, \infty[$

Analysis

Functions

Properties

- ▶ The set of all inputs to a particular function is called its **domain**
- ▶ The set of all outputs of a particular function is called its **image**
- ▶ The **codomain** is the set into which all of the output of the function is constrained to fall
- ▶ For every function its codomain **includes** its image.

Example

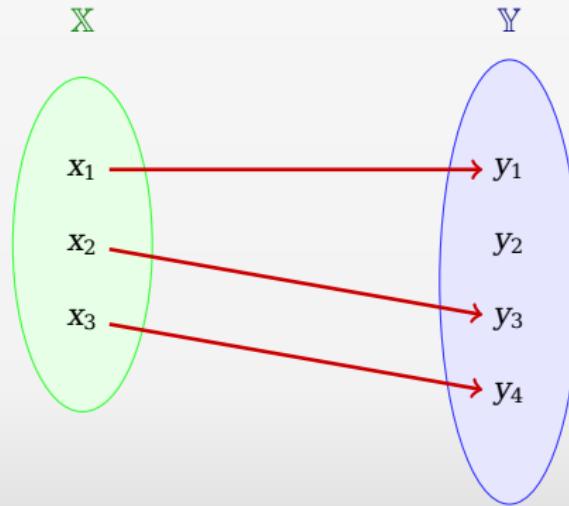
For a function $f : \mathbb{R} \mapsto \mathbb{R}$ defined by $f(x) = x^2$ the **codomain** of f is \mathbb{R} , but f does not map to any negative number. Thus the **image** of f , denoted $Im(f)$, is the set \mathbb{R}_0^+ that is the interval $[0, \infty[$

Analysis

One-to-one functions

Definition

A function $f : \mathbb{X} \mapsto \mathbb{Y}$ is a one-to-one function, or an injection, if each element in the codomain \mathbb{Y} is the image of at most one element in the domain \mathbb{X} .



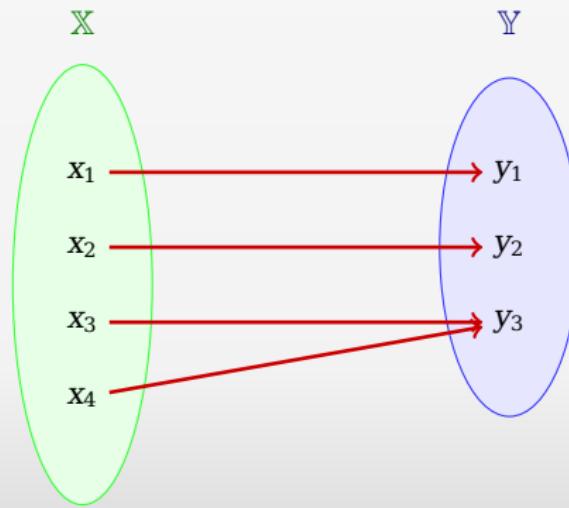
Analysis

Onto functions

Definition

A function $f : X \rightarrow Y$ is an onto function, or a surjection, if for every $y \in Y$, there is an $x \in X$ with $f(x) = y$.

Equivalently, a function $f : X \rightarrow Y$ is an onto function if $\text{Im}(f) = Y$.



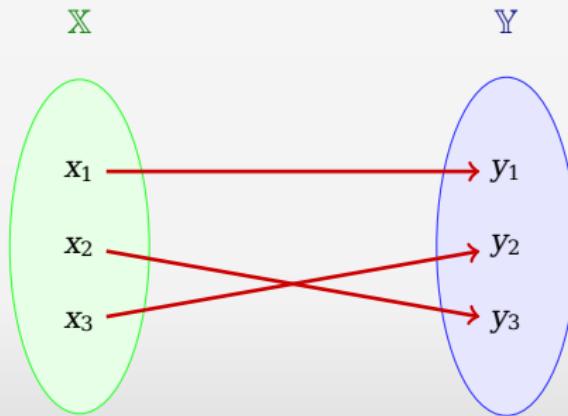
Analysis

Bijection

Definition

A function $f : \mathbb{X} \mapsto \mathbb{Y}$ is a bijection if and only if it satisfies the condition for every $y \in \mathbb{Y}$ there is a unique $x \in \mathbb{X}$ $f(x) = y$.

Equivalently, if a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is an injection and a surjection, then f is called a bijection.



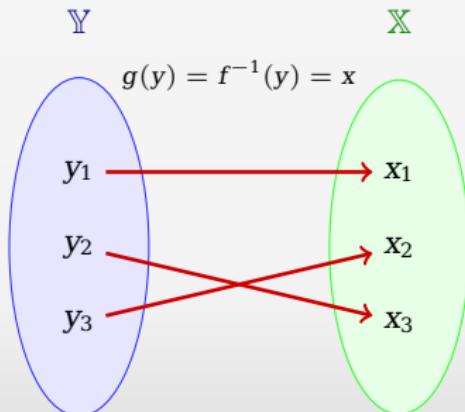
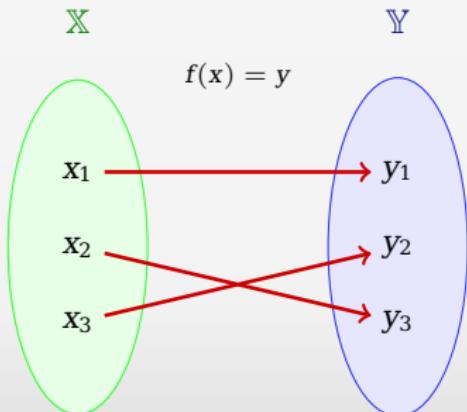
Analysis

Inverse function

Definition

If $f : \mathbb{X} \rightarrow \mathbb{Y}$ is a bijection then it is a simple matter to define a bijection $g : \mathbb{Y} \rightarrow \mathbb{X}$ as follows: for each $y \in \mathbb{Y}$ define $g(y) = x$ where $x \in \mathbb{X}$ and $f(x) = y$.

This function g obtained from f is called the inverse function of f and is denoted by $g = f^{-1}$.



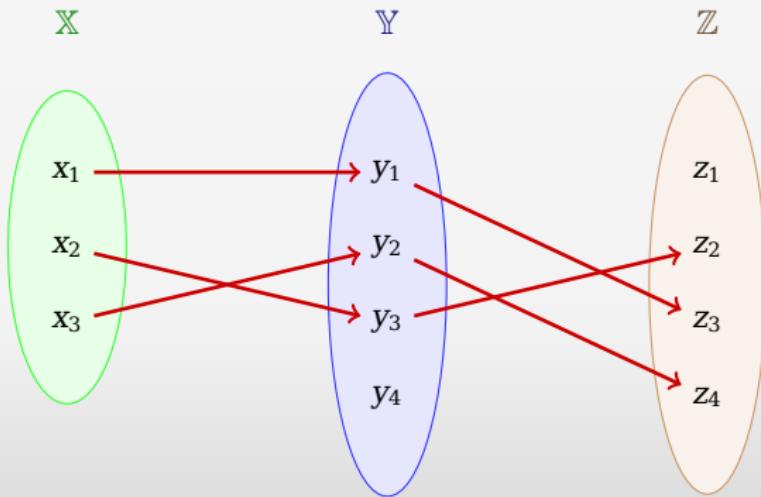
Analysis

Composition

Definition

The nesting of two or more functions to form a single new function is known as composition.

The composition of two functions $f : \mathbb{X} \mapsto \mathbb{Y}$ and $g : \mathbb{Y} \mapsto \mathbb{Z}$ is denoted $f \circ g$.
Composition is associative, so that $f \circ (g \circ h) = (f \circ g) \circ h$.



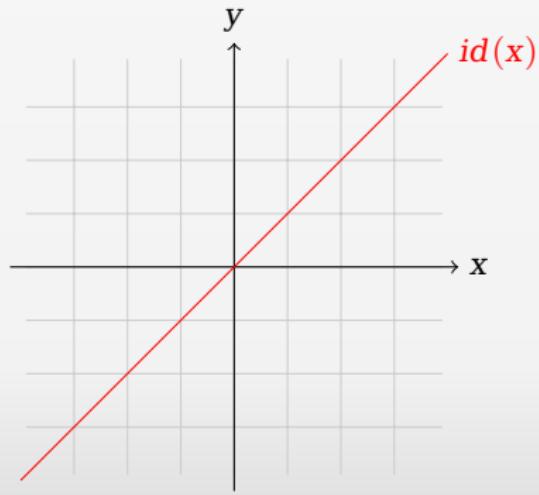
Analysis

Identity function

Definition

The unique function over a set \mathbb{X} that maps each element to itself is called the **identity** function for \mathbb{X} , and typically denoted by $id_{\mathbb{X}}$.

Each set has its own identity function.



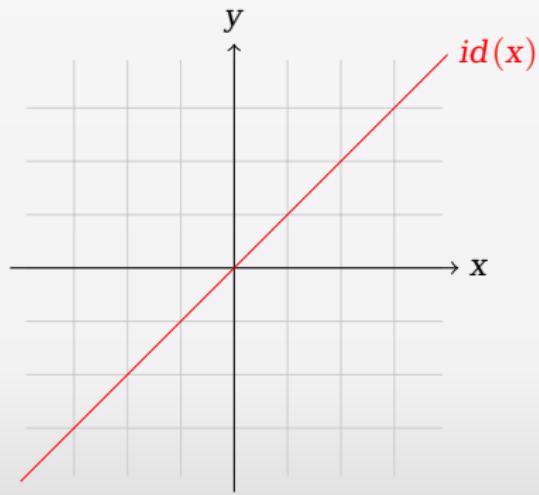
Analysis

Identity function

Definition

The unique function over a set \mathbb{X} that maps each element to itself is called the **identity** function for \mathbb{X} , and typically denoted by $id_{\mathbb{X}}$.

- ▶ Each set has its own identity function.
- ▶ Under composition, an identity function is neutral: if $f : \mathbb{X} \mapsto \mathbb{Y}$ is any function, then $f \circ id_{\mathbb{X}} = f$ and $id_{\mathbb{Y}} \circ f = f$.



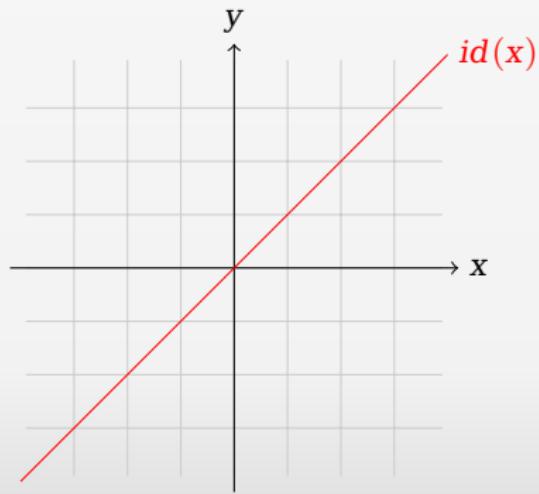
Analysis

Identity function

Definition

The unique function over a set \mathbb{X} that maps each element to itself is called the **identity** function for \mathbb{X} , and typically denoted by $id_{\mathbb{X}}$.

- ▶ Each set has its own identity function.
- ▶ Under composition, an identity function is neutral: if $f : \mathbb{X} \mapsto \mathbb{Y}$ is any function, then $f \circ id_{\mathbb{X}} = f$ and $id_{\mathbb{Y}} \circ f = f$.



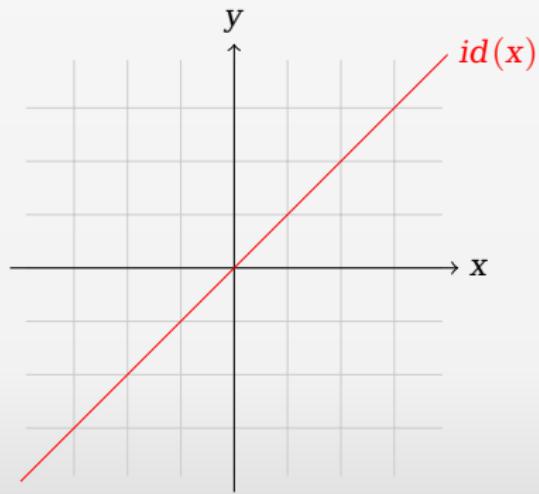
Analysis

Identity function

Definition

The unique function over a set \mathbb{X} that maps each element to itself is called the **identity** function for \mathbb{X} , and typically denoted by $id_{\mathbb{X}}$.

- ▶ Each set has its own identity function.
- ▶ Under composition, an identity function is neutral: if $f : \mathbb{X} \mapsto \mathbb{Y}$ is any function, then $f \circ id_{\mathbb{X}} = f$ and $id_{\mathbb{Y}} \circ f = f$.



Analysis

One-way functions

One-way functions : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is called a one-way function if:

- ▶ $f(x)$ is easy to compute $\forall x \in \mathbb{X}$ and,
- ▶ for essentially all elements $y \in \text{Im}(f)$ it is computationally infeasible to find $x \in \mathbb{X}$ such that $f(x) = y$.



Example

Analysis

One-way functions

One-way functions : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is called a one-way function if:

- ▶ $f(x)$ is easy to compute $\forall x \in \mathbb{X}$ and,
- ▶ for essentially all elements $y \in \text{Im}(f)$ it is computationally infeasible to find $x \in \mathbb{X}$ such that $f(x) = y$.



Example

• $f(x) = x^2$ is not one-way because it is easy to compute and given $y = x^2$ it is easy to find $x = \sqrt{y}$.

Analysis

One-way functions

One-way functions : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is called a one-way function if:

- ▶ $f(x)$ is easy to compute $\forall x \in \mathbb{X}$ and,
- ▶ for essentially all elements $y \in Im(f)$ it is computationally infeasible to find $x \in \mathbb{X}$ such that $f(x) = y$.



Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53893$, we have $n = p * q = 2624653723$.
- ▶ Let $\mathbb{Z} = \{1, 2, 3, \dots, n-1\}$ and define a function $f : \mathbb{Z} \rightarrow \mathbb{Z}$ by $f(x) = x^2 \pmod{n}$.

It is easy to calculate $f(x)$ for any $x \in \mathbb{Z}$, but it is computationally infeasible to find x given $y = f(x)$.

Analysis

One-way functions

One-way functions : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is called a **one-way** function if:

- ▶ $f(x)$ is easy to compute $\forall x \in \mathbb{X}$ and,
- ▶ for essentially all elements $y \in \text{Im}(f)$ it is computationally infeasible to find $x \in \mathbb{X}$ such that $f(x) = y$.



Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $\mathbb{X} = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : \mathbb{X} \mapsto \mathbb{X}$ by $f(x) = x^3 \bmod (n)$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \bmod (n) = 1981394214$.
- ▶ Computing $f(x)$ is a relatively simple thing to do but to reverse the procedure is much more difficult.

Analysis

One-way functions

One-way functions : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is called a **one-way** function if:

- ▶ $f(x)$ is easy to compute $\forall x \in \mathbb{X}$ and,
- ▶ for essentially all elements $y \in \text{Im}(f)$ it is computationally infeasible to find $x \in \mathbb{X}$ such that $f(x) = y$.



Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $\mathbb{X} = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : \mathbb{X} \mapsto \mathbb{X}$ by $f(x) = x^3 \pmod{n}$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \pmod{n} = 1981394214$.
- ▶ Computing $f(x)$ is a relatively simple thing to do but to reverse the procedure is much more difficult.

Analysis

One-way functions

One-way functions : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is called a one-way function if:

- ▶ $f(x)$ is easy to compute $\forall x \in \mathbb{X}$ and,
- ▶ for essentially all elements $y \in \text{Im}(f)$ it is computationally infeasible to find $x \in \mathbb{X}$ such that $f(x) = y$.



Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $X = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : X \mapsto X$ by $f(x) = x^3 \bmod (n)$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \bmod (n) = 1981394214$.
- ▶ Computing $f(x)$ is a relatively simple thing to do but to reverse the procedure is much more difficult.

Analysis

One-way functions

One-way functions : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is called a one-way function if:

- ▶ $f(x)$ is easy to compute $\forall x \in \mathbb{X}$ and,
- ▶ for essentially all elements $y \in \text{Im}(f)$ it is computationally infeasible to find $x \in \mathbb{X}$ such that $f(x) = y$.



Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $X = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : X \mapsto X$ by $f(x) = x^3 \bmod (n)$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \bmod (n) = 1981394214$.
- ▶ Computing $f(x)$ is a relatively simple thing to do but to reverse the procedure is much more difficult.

Analysis

One-way functions

One-way functions : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is called a one-way function if:

- ▶ $f(x)$ is easy to compute $\forall x \in \mathbb{X}$ and,
- ▶ for essentially all elements $y \in \text{Im}(f)$ it is computationally infeasible to find $x \in \mathbb{X}$ such that $f(x) = y$.



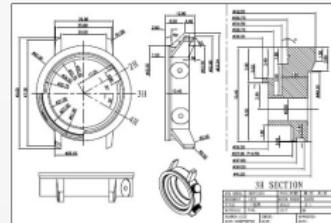
Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $X = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : X \mapsto X$ by $f(x) = x^3 \bmod (n)$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \bmod (n) = 1981394214$.
- ▶ Computing $f(x)$ is a relatively simple thing to do but to reverse the procedure is much more difficult.

Analysis

Trapdoor one-way functions

Trapdoor one-way function : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is a trapdoor one-way function if, knowing additionnal elements, it becomes easily feasible to find for any given $y \in \text{Im}(f)$, an $x \in \mathbb{X}$ such that $f(x) = y$.



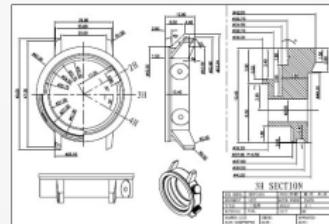
Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $X = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : X \mapsto X$ by $f(x) = x^3 \pmod{n}$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \pmod{n} = 1981394214$.
- ▶ With the additional information of the factors of $n = 2624653723$ – namely $p = 48611$ and $q = 53993$ – it becomes much easier to invert the function.

Analysis

Trapdoor one-way functions

Trapdoor one-way function : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is a trapdoor one-way function if, knowing additionnal elements, it becomes easily feasible to find for any given $y \in \text{Im}(f)$, an $x \in \mathbb{X}$ such that $f(x) = y$.



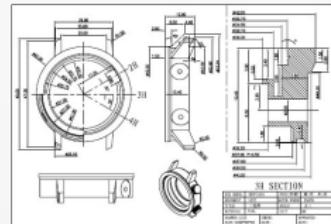
Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $X = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : X \mapsto X$ by $f(x) = x^3 \pmod{n}$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \pmod{n} = 1981394214$.
- ▶ With the additional information of the factors of $n = 2624653723$ – namely $p = 48611$ and $q = 53993$ – it becomes much easier to invert the function.

Analysis

Trapdoor one-way functions

Trapdoor one-way function : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is a trapdoor one-way function if, knowing additionnal elements, it becomes easily feasible to find for any given $y \in \text{Im}(f)$, an $x \in \mathbb{X}$ such that $f(x) = y$.



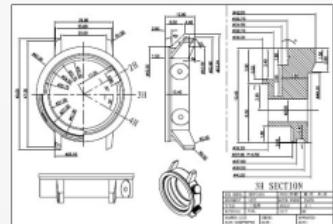
Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $X = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : X \mapsto X$ by $f(x) = x^3 \pmod{n}$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \pmod{n} = 1981394214$.
- ▶ With the additional information of the factors of $n = 2624653723$ – namely $p = 48611$ and $q = 53993$ – it becomes much easier to invert the function.

Analysis

Trapdoor one-way functions

Trapdoor one-way function : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is a trapdoor one-way function if, knowing additionnal elements, it becomes easily feasible to find for any given $y \in \text{Im}(f)$, an $x \in \mathbb{X}$ such that $f(x) = y$.



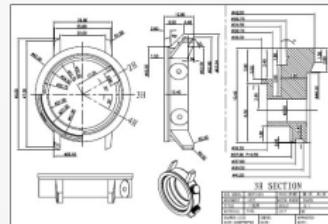
Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $X = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : X \mapsto X$ by $f(x) = x^3 \pmod{n}$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \pmod{n} = 1981394214$.
- ▶ With the additional information of the factors of $n = 2624653723$ – namely $p = 48611$ and $q = 53993$ – it becomes much easier to invert the function.

Analysis

Trapdoor one-way functions

Trapdoor one-way function : a function $f : \mathbb{X} \mapsto \mathbb{Y}$ is a trapdoor one-way function if, knowing additionnal elements, it becomes easily feasible to find for any given $y \in \text{Im}(f)$, an $x \in \mathbb{X}$ such that $f(x) = y$.



Example

- ▶ Select two prime numbers $p = 48611$ and $q = 53993$, we have $n = p * q = 2624653723$.
- ▶ Let $X = \{1, 2, 3, \dots, n - 1\}$ and define a function $f : X \mapsto X$ by $f(x) = x^3 \pmod{n}$.
- ▶ For instance, let $x = 2489991$, we have $f(x) = x^3 \pmod{n} = 1981394214$.
- ▶ With the additional information of the factors of $n = 2624653723$ – namely $p = 48611$ and $q = 53993$ – it becomes much easier to invert the function.

Analysis

Permutation

Permutations are functions which are often used in various cryptographic constructs

Definition

A **permutation** of a non empty set P is a bijective mapping $P \mapsto P$

Properties

- The set of permutations of a non empty set under the operation of composition, forms a group known as the symmetric group of P and denoted by S_P .

Analysis

Permutation

Permutations are functions which are often used in various cryptographic constructs

Definition

A **permutation** of a non empty set P is a bijective mapping $P \mapsto P$

Properties

- ▶ The **set of permutations** of P , under the operation of compositions, forms a group known as the **symmetric group** of P and denoted \mathfrak{S}_P
- ▶ If P is finite and with cardinality $n > 0, n \in \mathbb{N}$, the symmetric group of this set P is known as the **symmetric group of indice n** , it is denoted \mathfrak{S}_n
- ▶ The order of \mathfrak{S}_n is $n!$
- ▶ The elements of \mathfrak{S}_n are permutations
- ▶ An element of \mathfrak{S}_n that permutes two elements of n and leaves the other elements fixed is called a transposition

Analysis

Permutation

Permutations are functions which are often used in various cryptographic constructs

Definition

A **permutation** of a non empty set P is a bijective mapping $P \mapsto P$

Properties

- ▶ The **set of permutations** of P , under the operation of compositions, forms a group known as the **symmetric group** of P and denoted \mathfrak{S}_P
- ▶ If P is finite and with cardinality $n > 0, n \in \mathbb{N}$, the symmetric group of this set P is known as the **symmetric group of indice n** , it is denoted \mathfrak{S}_n
- ▶ The order of \mathfrak{S}_n is $n!$
- ▶ The elements of \mathfrak{S}_n are permutations
- ▶ An element of \mathfrak{S}_n that permutes two elements of n and leaves the other elements fixed is called a **transposition**

Analysis

Permutation

Permutations are functions which are often used in various cryptographic constructs

Definition

A **permutation** of a non empty set P is a bijective mapping $P \mapsto P$

Properties

- ▶ The **set of permutations** of P , under the operation of compositions, forms a group known as the **symmetric group** of P and denoted \mathfrak{S}_P
- ▶ If P is finite and with cardinality $n > 0, n \in \mathbb{N}$, the symmetric group of this set P is known as the **symmetric group of indice n** , it is denoted \mathfrak{S}_n
- ▶ The order of \mathfrak{S}_n is $n!$
- ▶ The elements of \mathfrak{S}_n are permutations
- ▶ An element of \mathfrak{S}_n that permutes two elements of P and leaves the other elements fixed is called a transposition

Analysis

Permutation

Permutations are functions which are often used in various cryptographic constructs

Definition

A **permutation** of a non empty set P is a bijective mapping $P \mapsto P$

Properties

- ▶ The **set of permutations** of P , under the operation of compositions, forms a group known as the **symmetric group** of P and denoted \mathfrak{S}_P
- ▶ If P is finite and with cardinality $n > 0, n \in \mathbb{N}$, the symmetric group of this set P is known as the **symmetric group of indice n** , it is denoted \mathfrak{S}_n
- ▶ The order of \mathfrak{S}_n is $n!$
- ▶ The elements of \mathfrak{S}_n are permutations
- ▶ An element of \mathfrak{S}_n that permutes two elements of P and leaves the other elements fixed is called a **transposition**

Analysis

Permutation

Permutations are functions which are often used in various cryptographic constructs

Definition

A **permutation** of a non empty set P is a bijective mapping $P \mapsto P$

Properties

- ▶ The **set of permutations** of P , under the operation of compositions, forms a group known as the **symmetric group** of P and denoted \mathfrak{S}_P
- ▶ If P is finite and with cardinality $n > 0, n \in \mathbb{N}$, the symmetric group of this set P is known as the **symmetric group of indice n** , it is denoted \mathfrak{S}_n
- ▶ The order of \mathfrak{S}_n is $n!$
- ▶ The elements of \mathfrak{S}_n are permutations
- ▶ An element of \mathfrak{S}_n that permutes two elements of P and leaves the other elements fixed is called a **transposition**

Analysis

Permutation

Permutations are functions which are often used in various cryptographic constructs

Definition

A **permutation** of a non empty set P is a bijective mapping $P \mapsto P$

Properties

- ▶ The **set of permutations** of P , under the operation of compositions, forms a group known as the **symmetric group** of P and denoted \mathfrak{S}_P
- ▶ If P is finite and with cardinality $n > 0, n \in \mathbb{N}$, the symmetric group of this set P is known as the **symmetric group of indice n** , it is denoted \mathfrak{S}_n
- ▶ The order of \mathfrak{S}_n is $n!$
- ▶ The elements of \mathfrak{S}_n are permutations
- ▶ An element of \mathfrak{S}_n that permutes two elements of P and leaves the other elements fixed is called a **transposition**

Analysis

Permutation

Example

- ▶ Let $P = \{1, 2, 3, 4, 5\}$
- ▶ A permutation $p : P \mapsto P$ is defined as follows:

$$p(1) = 3, p(2) = 5, p(3) = 4, p(4) = 2, p(5) = 1$$

- ▶ A permutation can be described as an array: $p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$
- ▶ Since permutations are bijections they have inverses: $p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 1 & 3 & 2 \end{pmatrix}$

Analysis

Permutation

Example

- ▶ Let $P = \{1, 2, 3, 4, 5\}$
- ▶ A permutation $p : P \mapsto P$ is defined as follows:

$$p(1) = 3, p(2) = 5, p(3) = 4, p(4) = 2, p(5) = 1$$

- ▶ A permutation can be described as an array: $p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$
- ▶ Since permutations are bijections they have inverses: $p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 1 & 3 & 2 \end{pmatrix}$

Analysis

Permutation

Example

- ▶ Let $P = \{1, 2, 3, 4, 5\}$
- ▶ A permutation $p : P \mapsto P$ is defined as follows:

$$p(1) = 3, p(2) = 5, p(3) = 4, p(4) = 2, p(5) = 1$$

- ▶ A permutation can be described as an array: $p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$
- ▶ Since permutations are bijections they have inverses: $p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 1 & 3 & 2 \end{pmatrix}$

Analysis

Permutation

Example

- ▶ Let $P = \{1, 2, 3, 4, 5\}$
- ▶ A permutation $p : P \mapsto P$ is defined as follows:

$$p(1) = 3, p(2) = 5, p(3) = 4, p(4) = 2, p(5) = 1$$

- ▶ A permutation can be described as an array: $p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$
- ▶ Since permutations are bijections they have inverses: $p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 1 & 3 & 2 \end{pmatrix}$

Analysis

Permutation

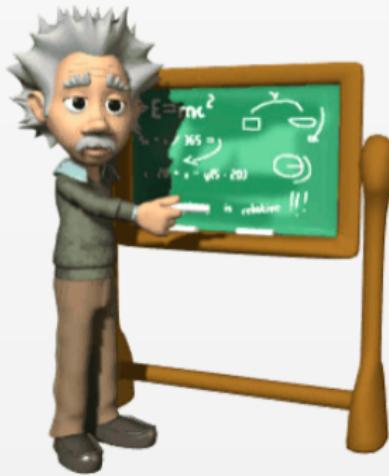
Example

- ▶ Let $P = \{1, 2, 3, 4, 5\}$
- ▶ A permutation $p : P \mapsto P$ is defined as follows:

$$p(1) = 3, p(2) = 5, p(3) = 4, p(4) = 2, p(5) = 1$$

- ▶ A permutation can be described as an array: $p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$
- ▶ Since permutations are bijections they have inverses: $p^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 1 & 3 & 2 \end{pmatrix}$

Abstract algebra



Abstract algebra

Definition

Abstract algebra is the subject area of mathematics that **studies algebraic structures** such as groups, rings, fields, modules, vector spaces, and algebras.

Contemporary cryptography makes extensive use of abstract algebra.



Abstract algebra

The **sets**

Definition

- ▶ A set S is a collection of distinct objects

Properties

A set has three properties:

- it contains elements
- it is well defined
- it is unique

Abstract algebra

The **sets**

Definition

- ▶ A set **S** is a collection of distinct objects

Properties

- ▶ A set is described by giving a common description of its elements or by listing an extended definition (e.g. $\{1, 2, 3\}$)

Abstract algebra

The sets

Definition

- ▶ A set S is a collection of distinct objects

Properties

- ▶ A set is described by using a semantic description: *the set of integers* or by using an extensional definition: $S = \{2, 4, 6, 8\}$
- ▶ The key relation between sets is membership, if s is a member of S , this is denoted $s \in S$
- ▶ If every member of set A is also a member of set B , then A is said to be a subset of B , written $A \subset B$
- ▶ The cardinality $|S|$ of a set S is the number of members of S . Thus, if $S = \{1, 2, 3, 4, 5\}$, then $|S| = 5$.
- ▶ There is a unique set with no members, which is called the empty set and is denoted \emptyset .
- ▶ Some sets have infinite cardinality. The set \mathbb{N} of natural numbers is infinite.

Abstract algebra

The sets

Definition

- ▶ A set S is a collection of distinct objects

Properties

- ▶ A set is describing by using a semantic description: *the set of integers* or by using an extentional definition: $S = \{2, 4, 6, 8\}$
- ▶ The key relation between sets is membership, if s is a member of S , this is denoted $s \in S$
- ▶ If every member of set A is also a member of set B , then A is said to be a subset of B , written $A \subset B$
- ▶ The cardinality $|S|$ of a set S is the number of members of S . Thus, if $S = \{2, 4, 6, 8\}$ then $|S| = 4$.
- ▶ There is a unique set with no members, which is called the empty set and is denoted
- ▶ Some sets have infinite cardinality. The set \mathbb{N} of natural numbers is infinite.

Abstract algebra

The sets

Definition

- ▶ A set S is a collection of distinct objects

Properties

- ▶ A set is describing by using a semantic description: *the set of integers* or by using an extentional definition: $S = \{2, 4, 6, 8\}$
- ▶ The key relation between sets is membership, if s is a member of S , this is denoted $s \in S$
- ▶ If every member of set A is also a member of set B , then A is said to be a subset of B , written $A \subset B$
- ▶ The cardinality $|S|$ of a set S is the number of members of S . Thus, if $S = \{2, 4, 6, 8\}$ then $|S| = 4$.
- ▶ There is a unique set with no members, which is called the empty set and is denoted \varnothing
- ▶ Some sets have infinite cardinality. The set \mathbb{N} of natural numbers is infinite.

Abstract algebra

The sets

Definition

- ▶ A set S is a collection of distinct objects

Properties

- ▶ A set is describing by using a semantic description: *the set of integers* or by using an extentional definition: $S = \{2, 4, 6, 8\}$
- ▶ The key relation between sets is membership, if s is a member of S , this is denoted $s \in S$
- ▶ If every member of set A is also a member of set B , then A is said to be a subset of B , written $A \subset B$
- ▶ The cardinality $|S|$ of a set S is the number of members of S . Thus, if $S = \{2, 4, 6, 8\}$ then $|S| = 4$.
- ▶ There is a unique set with no members, which is called the empty set and is denoted \emptyset
- ▶ Some sets have infinite cardinality. The set \mathbb{N} of natural numbers is infinite.

Abstract algebra

The sets

Definition

- ▶ A set S is a collection of distinct objects

Properties

- ▶ A set is describing by using a semantic description: *the set of integers* or by using an extentional definition: $S = \{2, 4, 6, 8\}$
- ▶ The key relation between sets is membership, if s is a member of S , this is denoted $s \in S$
- ▶ If every member of set A is also a member of set B , then A is said to be a subset of B , written $A \subset B$
- ▶ The cardinality $|S|$ of a set S is the number of members of S . Thus, if $S = \{2, 4, 6, 8\}$ then $|S| = 4$.
- ▶ There is a unique set with no members, which is called the empty set and is denoted \emptyset
- ▶ Some sets have infinite cardinality. The set \mathbb{N} of natural numbers is infinite.

Abstract algebra

The sets

Definition

- ▶ A set S is a collection of distinct objects

Properties

- ▶ A set is described by using a semantic description: *the set of integers* or by using an extensional definition: $S = \{2, 4, 6, 8\}$
- ▶ The key relation between sets is membership, if s is a member of S , this is denoted $s \in S$
- ▶ If every member of set A is also a member of set B , then A is said to be a subset of B , written $A \subset B$
- ▶ The cardinality $|S|$ of a set S is the number of members of S . Thus, if $S = \{2, 4, 6, 8\}$ then $|S| = 4$.
- ▶ There is a unique set with no members, which is called the empty set and is denoted \emptyset
- ▶ Some sets have infinite cardinality. The set \mathbb{N} of natural numbers is infinite.

Abstract algebra

The sets

Definition

- ▶ A set S is a collection of distinct objects

Properties

- ▶ A set is describing by using a semantic description: *the set of integers* or by using an extentional definition: $S = \{2, 4, 6, 8\}$
- ▶ The key relation between sets is membership, if s is a member of S , this is denoted $s \in S$
- ▶ If every member of set A is also a member of set B , then A is said to be a subset of B , written $A \subset B$
- ▶ The cardinality $|S|$ of a set S is the number of members of S . Thus, if $S = \{2, 4, 6, 8\}$ then $|S| = 4$.
- ▶ There is a unique set with no members, which is called the empty set and is denoted \emptyset
- ▶ Some sets have infinite cardinality. The set \mathbb{N} of natural numbers is infinite.

Abstract algebra

The groups

The **groups** constitute the algebraic basic structure in mathematics since from them, the rings, the fields and the vector spaces are formed.

Definition

Let G a non empty set with an internal composition law $\bullet : G \times G \mapsto G$.

So G is a group, denoted (G, \bullet) , if:

- ▶ the internal composition law is associative
- ▶ $\forall g \in G$ it exists a neutral element $e \in G$ such that $e \bullet g = g \bullet e = g$
- ▶ e is called the identity element
- ▶ $\forall g \in G$ it exists a unique $g^{-1} \in G$ such that $g \bullet g^{-1} = g^{-1} \bullet g = e$

Abstract algebra

The groups

The **groups** constitute the algebraic basic structure in mathematics since from them, the rings, the fields and the vector spaces are formed.

Definition

Let G a non empty set with an internal composition law $\bullet : G \times G \mapsto G$.

So G is a group, denoted (G, \bullet) , if:

- ▶ the internal composition law is associative
- ▶ $\forall g \in G$ it exists a neutral element $e \in G$ such that $e \bullet g = g \bullet e = g$
- ▶ e is called the identity element
- ▶ $\forall g \in G$ it exists a unique $g^{-1} \in G$ such that $g \bullet g^{-1} = g^{-1} \bullet g = e$

Abstract algebra

The groups

The **groups** constitute the algebraic basic structure in mathematics since from them, the rings, the fields and the vector spaces are formed.

Definition

Let G a non empty set with an internal composition law $\bullet : G \times G \mapsto G$.

So G is a group, denoted (G, \bullet) , if:

- ▶ the internal composition law is associative
- ▶ $\forall g \in G$ it exists a neutral element $e \in G$ such that $e \bullet g = g \bullet e = g$
- ▶ e is called the identity element
- ▶ $\forall g \in G$ it exists a unique $g^{-1} \in G$ such that $g \bullet g^{-1} = g^{-1} \bullet g = e$

Abstract algebra

The groups

The **groups** constitute the algebraic basic structure in mathematics since from them, the rings, the fields and the vector spaces are formed.

Definition

Let G a non empty set with an internal composition law $\bullet : G \times G \mapsto G$.

So G is a group, denoted (G, \bullet) , if:

- ▶ the internal composition law is associative
- ▶ $\forall g \in G$ it exists a neutral element $e \in G$ such that $e \bullet g = g \bullet e = g$
- ▶ e is called the identity element
- ▶ $\forall g \in G$ it exists a unique $g^{-1} \in G$ such that $g \bullet g^{-1} = g^{-1} \bullet g = e$

Abstract algebra

The groups

The **groups** constitute the algebraic basic structure in mathematics since from them, the rings, the fields and the vector spaces are formed.

Definition

Let G a non empty set with an internal composition law $\bullet : G \times G \mapsto G$.

So G is a group, denoted (G, \bullet) , if:

- ▶ the internal composition law is associative
- ▶ $\forall g \in G$ it exists a neutral element $e \in G$ such that $e \bullet g = g \bullet e = g$
- ▶ e is called the identity element
- ▶ $\forall g \in G$ it exists a unique $g^{-1} \in G$ such that $g \bullet g^{-1} = g^{-1} \bullet g = e$

Abstract algebra

The groups

Properties

- ▶ The order of (G, \bullet) is the cardinality of the set G
- ▶ If the order of (G, \bullet) is finite then (G, \bullet) is a finite group
- ▶ A group whose the internal composition law is commutative is a **commutative group** so called an **abelian group**

Example of group

The set of integers \mathbb{Z} with the addition operation forms an abelian group denoted $(\mathbb{Z}, +)$.

Abstract algebra

The groups

Properties

- The order of (G, \bullet) is the cardinality of the set G
- If the order of (G, \bullet) is finite then (G, \bullet) is a finite group
- A group whose the internal composition law is commutative is a **commutative group** so called an **abelian group**

Example of group

The set of integers \mathbb{Z} with the **addition** operation forms an abelian group denoted $(\mathbb{Z}, +)$.

Abstract algebra

The groups

Properties

- ▶ The order of (G, \bullet) is the cardinality of the set G
- ▶ If the order of (G, \bullet) is finite then (G, \bullet) is a finite group
- ▶ A group whose the internal composition law is commutative is a **commutative group** so called an **abelian group**

Example of group

The set of integers \mathbb{Z} with the **addition** operation forms an abelian group denoted $(\mathbb{Z}, +)$.

Abstract algebra

The groups

Properties

- ▶ The order of (G, \bullet) is the cardinality of the set G
- ▶ If the order of (G, \bullet) is finite then (G, \bullet) is a finite group
- ▶ A group whose the internal composition law is commutative is a **commutative group** so called an **abelian group**

Example of group

The set of integers \mathbb{Z} with the **addition** operation forms an abelian group denoted $(\mathbb{Z}, +)$.

Abstract algebra

The groups

Properties

- ▶ The order of (G, \bullet) is the cardinality of the set G
- ▶ If the order of (G, \bullet) is finite then (G, \bullet) is a finite group
- ▶ A group whose the internal composition law is commutative is a **commutative group** so called an **abelian group**

Example of group

The set of integers \mathbb{Z} with the **addition** operation forms an abelian group denoted $(\mathbb{Z}, +)$.

Abstract algebra

The groups

Set: E

$\bullet : E \times E \mapsto E$

Abelian group: (E, \bullet)

Group: (E, \bullet)

associative $\forall x, y, z \in E$
 $(x \bullet y) \bullet z = x \bullet (y \bullet z)$

neutral element $\forall x \in E, \exists e \in E$
 $e \bullet x = x \bullet e = x$

inverse $\forall x \in E, \exists x^{-1} \in E$
 $x \bullet x^{-1} = x^{-1} \bullet x = e$

commutative $\forall x, y \in E$
 $x \bullet y = y \bullet x$

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
- ▶ the operation \bullet is associative
- ▶ the operation \bullet is distributive over $+$
- ▶ there is an element $0 \in A$ such that

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
- ▶ the operation \bullet is associative
- ▶ the operation \bullet is distributive over $+$
- ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a \forall a \in A$

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \cdot)$.

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
- ▶ the operation \bullet is associative
- ▶ the operation \bullet is distributive over $+$
- ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a, \forall a \in A$

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
- ▶ the operation \bullet is associative
- ▶ the operation \bullet is distributive over $+$
- ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a, \forall a \in A$

Some properties of rings:
- Every ring has a unique additive identity (0).
- Every ring has a unique multiplicative identity (1).
- Every ring has additive inverses.
- Every ring has a multiplicative inverse for every non-zero element.

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
- ▶ the operation \bullet is associative
- ▶ the operation \bullet is distributive over $+$
- ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a, \forall a \in A$

→ The following definition of the group can also be used to the definition of ring.

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The rings

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
- ▶ the operation \bullet is associative
- ▶ the operation \bullet is distributive over $+$
- ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a, \forall a \in A$

- ▶ The identity element of the group $(A, +)$ is 0 and is the zero of the ring $(A, +, \bullet)$
- ▶ The element 1 is the identity of the ring $(A, +, \bullet)$
- ▶ A ring is commutative if its second law is commutative

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
 - ▶ the operation \bullet is associative
 - ▶ the operation \bullet is distributive over $+$
 - ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a, \forall a \in A$
-
- ▶ The identity element of the group $(A, +)$ is 0 and is the zero of the ring $(A, +, \bullet)$
 - ▶ The element 1 is the identity of the ring $(A, +, \bullet)$
 - ▶ A ring is **commutative** if its second law is commutative

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
 - ▶ the operation \bullet is associative
 - ▶ the operation \bullet is distributive over $+$
 - ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a, \forall a \in A$
-
- ▶ The identity element of the group $(A, +)$ is 0 and is the zero of the ring $(A, +, \bullet)$
 - ▶ The element 1 is the identity of the ring $(A, +, \bullet)$
 - ▶ A ring is **commutative** if its second law is commutative

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
 - ▶ the operation \bullet is associative
 - ▶ the operation \bullet is distributive over $+$
 - ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a, \forall a \in A$
-
- ▶ The identity element of the group $(A, +)$ is 0 and is the zero of the ring $(A, +, \bullet)$
 - ▶ The element 1 is the identity of the ring $(A, +, \bullet)$
 - ▶ A ring is **commutative** if its second law is commutative

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The **rings**

Definition

Let A a non empty set with two binary operations $+$ and \bullet with $A \times A \mapsto A$.

Then $(A, +, \bullet)$ is a ring if:

- ▶ $(A, +)$ is an abelian group
 - ▶ the operation \bullet is associative
 - ▶ the operation \bullet is distributive over $+$
 - ▶ there is an element $1 \in A$ such that $1 \bullet a = a \bullet 1 = a, \forall a \in A$
-
- ▶ The identity element of the group $(A, +)$ is 0 and is the zero of the ring $(A, +, \bullet)$
 - ▶ The element 1 is the identity of the ring $(A, +, \bullet)$
 - ▶ A ring is **commutative** if its second law is commutative

Example of ring

The set of integers \mathbb{Z} under the operations of addition and multiplication forms a commutative ring denoted $(\mathbb{Z}, +, \bullet)$.

Abstract algebra

The **rings**

Set: E

$+ : E \times E \mapsto E$

$\bullet : E \times E \mapsto E$

Commutative ring: $(E, +, \bullet)$

Ring: $(E, +, \bullet)$

Abelian group: $(E, +)$

Group: $(E, +)$

associative $\forall x, y, z \in E$
 $(x + y) + z = x + (y + z)$

neutral element $\forall x \in E, \exists 0 \in E$
 $0 + x = x + 0 = x$

inverse $\forall x \in E, \exists -x \in E$
 $x + (-x) = (-x) + x = 0$

commutative $\forall x, y \in E$
 $x + y = y + x$

distributive $\forall x, y, z \in E$
 $x \bullet (y + z) = (x \bullet y) + (x \bullet z)$
 $(y + z) \bullet x = (y \bullet x) + (z \bullet y)$

associative $\forall x, y, z \in E$
 $(x \bullet y) \bullet z = x \bullet (y \bullet z)$

neutral element $\forall x \neq 0 \in E, \exists 1 \in E$
 $1 \bullet x = x \bullet 1 = x$

inverse $\forall x \in E, \exists x^{-1} \in E$
 $x \bullet x^{-1} = x^{-1} \bullet x = 1$

commutative $\forall x, y \in E$
 $x \bullet y = y \bullet x$

Abstract algebra

The fields

Definition

- ▶ A field is a ring in which every non empty element is invertible
- ▶ (*other definition*) – A field C is the ring $(C, +, \bullet)$ such that $(C, +)$ and $(C \setminus \{0\}, \bullet)$ are abelian groups
- ▶ The order of a field is the number of elements that it contains

Example of field

The set of rational numbers \mathbb{Q} , the set of real numbers \mathbb{R} and the set of complex numbers \mathbb{C} are fields under the operations of addition and multiplication.

Abstract algebra

The fields

Definition

- ▶ A field is a ring in which every non empty element is invertible
- ▶ (*other definition*) – A field C is the ring $(C, +, \bullet)$ such that $(C, +)$ and $(C \setminus \{0\}, \bullet)$ are abelian groups
- ▶ The order of a field is the number of elements that it contains

Example of field

The set of rational numbers \mathbb{Q} , the set of real numbers \mathbb{R} and the set of complex numbers \mathbb{C} are fields under the operations of addition and multiplication.

Abstract algebra

The fields

Definition

- ▶ A field is a ring in which every non empty element is invertible
- ▶ (*other definition*) – A field C is the ring $(C, +, \bullet)$ such that $(C, +)$ and $(C \setminus \{0\}, \bullet)$ are abelian groups
- ▶ The order of a field is the number of elements that it contains

Example of field

The set of rational numbers \mathbb{Q} , the set of real numbers \mathbb{R} and the set of complex numbers \mathbb{C} are fields under the operations of addition and multiplication.

Abstract algebra

The fields

Definition

- ▶ A field is a ring in which every non empty element is invertible
- ▶ (*other definition*) – A field C is the ring $(C, +, \bullet)$ such that $(C, +)$ and $(C \setminus \{0\}, \bullet)$ are abelian groups
- ▶ The order of a field is the number of elements that it contains

Example of field

The set of rational numbers \mathbb{Q} , the set of real numbers \mathbb{R} and the set of complex numbers \mathbb{C} are fields under the operations of addition and multiplication.

Abstract algebra

The fields

Definition

- ▶ A field is a ring in which every non empty element is invertible
- ▶ (*other definition*) – A field C is the ring $(C, +, \bullet)$ such that $(C, +)$ and $(C \setminus \{0\}, \bullet)$ are abelian groups
- ▶ The order of a field is the number of elements that it contains

Example of field

The set of rational numbers \mathbb{Q} , the set of real numbers \mathbb{R} and the set of complex numbers \mathbb{C} are fields under the operations of addition and multiplication.

Abstract algebra

The finite fields or **Galois fields**

Definition

- ▶ The set $\mathbb{Z}_p = \{0, \dots, p - 1\}$, denoted \mathbb{Z}/p , with the operations of addition and multiplication defined modulo p forms a finite field if and only if p is prime
- ▶ This field is called Galois field of order p and is denoted $GF(p)$
- ▶ The number of elements of a Galois field, its **order**, is of the form p^n where p is a prime number called the **characteristic** of the field and n is a positive integer

Smallest Galois field

Abstract algebra

The finite fields or Galois fields

Definition

- ▶ The set $\mathbb{Z}_p = \{0, \dots, p - 1\}$, denoted \mathbb{Z}/p , with the operations of addition and multiplication defined modulo p forms a finite field if and only if p is prime
- ▶ This field is called Galois field of order p and is denoted $GF(p)$
- ▶ The number of elements of a Galois field, its *order*, is of the form p^n where p is a prime number called the *characteristic* of the field and n is a positive integer

Smallest Galois field

Abstract algebra

The finite fields or Galois fields

Definition

- ▶ The set $\mathbb{Z}_p = \{0, \dots, p - 1\}$, denoted \mathbb{Z}/p , with the operations of addition and multiplication defined modulo p forms a finite field if and only if p is prime
- ▶ This field is called Galois field of order p and is denoted $GF(p)$
- ▶ The number of elements of a Galois field, its *order*, is of the form p^n where p is a prime number called the *characteristic* of the field and n is a positive integer

Smallest Galois field

Abstract algebra

The finite fields or Galois fields

Definition

- ▶ The set $\mathbb{Z}_p = \{0, \dots, p - 1\}$, denoted \mathbb{Z}/p , with the operations of addition and multiplication defined modulo p forms a finite field if and only if p is prime
- ▶ This field is called Galois field of order p and is denoted $GF(p)$
- ▶ The number of elements of a Galois field, its **order**, is of the form p^n where p is a prime number called the **characteristic** of the field and n is a positive integer

Smallest Galois field

- ▶ The smallest finite field $GF(2)$ contains the 0 and 1 elements and we have $GF(2) = \{0, 1\}$.
- ▶ In $GF(2)$ the addition modulo 2 is realized by a XOR and the multiplication modulo 2 by an AND.

Abstract algebra

The finite fields or Galois fields

Definition

- ▶ The set $\mathbb{Z}_p = \{0, \dots, p - 1\}$, denoted \mathbb{Z}/p , with the operations of addition and multiplication defined modulo p forms a finite field if and only if p is prime
- ▶ This field is called Galois field of order p and is denoted $GF(p)$
- ▶ The number of elements of a Galois field, its **order**, is of the form p^n where p is a prime number called the **characteristic** of the field and n is a positive integer

Smallest Galois field

- ▶ The smallest finite field $GF(2)$ contains the 0 and 1 elements and we have $GF(2) = \{0, 1\}$.
- ▶ In $GF(2)$ the addition modulo 2 is realized by a XOR and the multiplication modulo 2 by an AND.

Abstract algebra

The finite fields or Galois fields

Definition

- ▶ The set $\mathbb{Z}_p = \{0, \dots, p - 1\}$, denoted \mathbb{Z}/p , with the operations of addition and multiplication defined modulo p forms a finite field if and only if p is prime
- ▶ This field is called Galois field of order p and is denoted $GF(p)$
- ▶ The number of elements of a Galois field, its **order**, is of the form p^n where p is a prime number called the **characteristic** of the field and n is a positive integer

Smallest Galois field

- ▶ The smallest finite field $GF(2)$ contains the **0** and **1** elements and we have $GF(2) = \{0, 1\}$.
- ▶ In $GF(2)$ the addition modulo 2 is realized by a **XOR** and the multiplication modulo 2 by an **AND**.

Abstract algebra

The finite fields or Galois fields

Definition

- ▶ The set $\mathbb{Z}_p = \{0, \dots, p - 1\}$, denoted \mathbb{Z}/p , with the operations of addition and multiplication defined modulo p forms a finite field if and only if p is prime
- ▶ This field is called Galois field of order p and is denoted $GF(p)$
- ▶ The number of elements of a Galois field, its **order**, is of the form p^n where p is a prime number called the **characteristic** of the field and n is a positive integer

Smallest Galois field

- ▶ The smallest finite field $GF(2)$ contains the 0 and 1 elements and we have $GF(2) = \{0, 1\}$.
- ▶ In $GF(2)$ the addition modulo 2 is realized by a **XOR** and the multiplication modulo 2 by an **AND**.

Abstract algebra

The finite fields or Galois fields

For every prime number p and positive integer n there exists a finite field with p^n elements denoted $GF(p^n)$

- ▶ For example the AES algorithm is construct over $GF(2^8)$
- ▶ The atomic element used by AES is the byte considered as an element of $GF(2^8)$ under the form of a polynomial with coefficients in $\{0, 1\}$:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

- ▶ Thus, the char 'a' has the hexadecimal value 61 in the correspondence table ASCII, be 01100001 in binary. Its polynomial translation is the next one:

$$x^6 + x^5 + 1$$

Abstract algebra

The finite fields or Galois fields

For every prime number p and positive integer n there exists a finite field with p^n elements denoted $GF(p^n)$

- ▶ For example the AES algorithm is construct over $GF(2^8)$
- ▶ The atomic element used by AES is the byte considered as an element of $GF(2^8)$ under the form of a polynomial with coefficients in $\{0, 1\}$:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

- ▶ Thus, the char 'a' has the hexadecimal value 61 in the correspondence table ASCII, be 01100001 in binary. Its polynomial translation is the next one:

$$x^6 + x^5 + 1$$

Abstract algebra

The finite fields or Galois fields

For every prime number p and positive integer n there exists a finite field with p^n elements denoted $GF(p^n)$

- ▶ For example the AES algorithm is construct over $GF(2^8)$
- ▶ The atomic element used by AES is the byte considered as an element of $GF(2^8)$ under the form of a polynomial with coefficients in $\{0, 1\}$:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

- ▶ Thus, the char 'a' has the hexadecimal value 61 in the correspondence table ASCII, be 01100001 in binary. Its polynomial translation is the next one:

$$x^6 + x^5 + 1$$

Abstract algebra

The finite fields or Galois fields

For every prime number p and positive integer n there exists a finite field with p^n elements denoted $GF(p^n)$

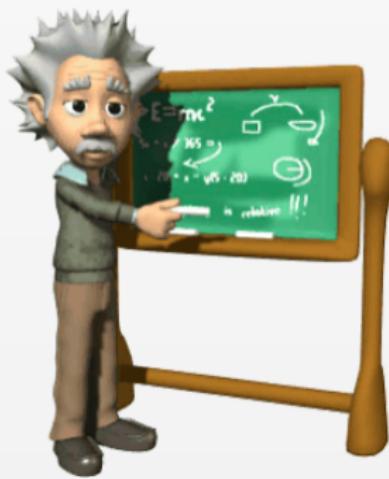
- ▶ For example the AES algorithm is construct over $GF(2^8)$
- ▶ The atomic element used by AES is the byte considered as an element of $GF(2^8)$ under the form of a polynomial with coefficients in $\{0, 1\}$:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

- ▶ Thus, the char 'a' has the hexadecimal value 61 in the correspondence table ASCII, be 01100001 in binary. Its polynomial translation is the next one:

$$x^6 + x^5 + 1$$

Number theory



Number theory

"Mathematics is the queen of sciences and number theory is the queen of mathematics."

Carl Friedrich Gauss

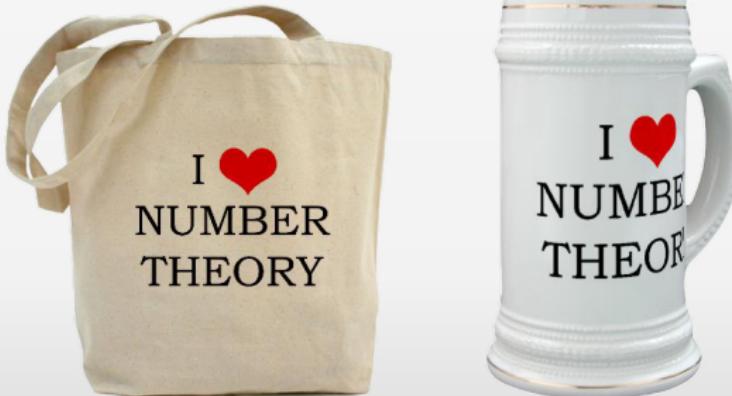


Number theory

Definition

Number theory is the branch of pure mathematics concerned with **the properties of numbers** in general, and **integers in particular**, as well as the wider classes of problems that arise from their study.

Particularly through the use of **prime numbers**, cryptography makes extensive use of number theory.



Number theory

Prime numbers

Definition

A prime number is a positive integer $p > 1$ that has no positive integer divisors other than 1 and p itself.

Properties

→ The property of being prime is called primality



Number theory

Prime numbers

Definition

A prime number is a positive integer $p > 1$ that has no positive integer divisors other than 1 and p itself.

Properties

- The property of being prime is called **primality**
- 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 et 97 are the prime numbers less than 100
- The largest known prime number is $2^{43,717,691} - 1$, it was discovered August 23, 2008. It's a Mersenne number called *M43*, it contains 12,972,131 digits



Number theory

Prime numbers

Definition

A prime number is a positive integer $p > 1$ that has no positive integer divisors other than 1 and p itself.

Properties

- ▶ The property of being prime is called **primality**
- ▶ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 et 97 are the prime numbers less than 100
- ▶ The largest known prime number is $2^{43112609} - 1$, it was discovered August 23, 2008. It's a **Mersenne** number called M_{47} , it contains 12978189 digits



Number theory

Prime numbers

Definition

A prime number is a positive integer $p > 1$ that has no positive integer divisors other than 1 and p itself.

Properties

- ▶ The property of being prime is called **primality**
- ▶ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 et 97 are the prime numbers less than 100
- ▶ The largest known prime number is $2^{43112609} - 1$, it was discovered August 23, 2008. It's a **Mersenne** number called M_{47} , it contains 12978189 digits



Number theory

Prime numbers

Definition

A prime number is a positive integer $p > 1$ that has no positive integer divisors other than 1 and p itself.

Properties

- ▶ The property of being prime is called **primality**
- ▶ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 et 97 are the prime numbers less than 100
- ▶ The largest known prime number is $2^{43112609} - 1$, it was discovered August 23, 2008. It's a **Mersenne** number called M_{47} , it contains 12978189 digits



Number theory

Prime numbers

Prime numbers in nature

- ▶ One example of the use of prime numbers in nature is as an evolutionary strategy used by **cicadas** of the genus **Magicicada**.
- ▶ These insects spend most of their lives as **grubs** underground.
- ▶ They only pupate and then emerge from their burrows after 13 or 17 years, at which point they fly about, breed, and then die after a few weeks at most.
- ▶ The logic for this is believed to be that the prime number intervals between emergences make it very difficult for predators to evolve that could specialize as predators on Magicicadas.
- ▶ If Magicicadas appeared at a non-prime number intervals, say every 12, 15, 18, ..., then predators appearing every 3, 5, 6, 9, or ... years would be sure to meet them.

Number theory

Prime numbers

Prime numbers in nature

- ▶ One example of the use of prime numbers in nature is as an evolutionary strategy used by **cicadas** of the genus **Magicicada**.
- ▶ These insects spend most of their lives as **grubs** underground.
- ▶ They only pupate and then emerge from their burrows after **13** or **17** years, at which point they fly about, breed, and then die after a few weeks at most.
- ▶ The logic for this is believed to be that the prime number intervals between emergences make it very difficult for predators to evolve that could specialize as predators on Magicicadas.
- ▶ If Magicicadas appeared at a non-prime number intervals, say every **12**, **18**, **24**, **30**, **36**, **42**, **48**, **54**, **60**, **66**, **72**, **78**, **84**, **90**, **96**, **102**, **108**, **114**, **120**, **126**, **132**, **138**, **144**, **150**, **156**, **162**, **168**, **174**, **180**, **186**, **192**, **198**, **204**, **210**, **216**, **222**, **228**, **234**, **240**, **246**, **252**, **258**, **264**, **270**, **276**, **282**, **288**, **294**, **300**, **306**, **312**, **318**, **324**, **330**, **336**, **342**, **348**, **354**, **360**, **366**, **372**, **378**, **384**, **390**, **396**, **402**, **408**, **414**, **420**, **426**, **432**, **438**, **444**, **450**, **456**, **462**, **468**, **474**, **480**, **486**, **492**, **498**, **504**, **510**, **516**, **522**, **528**, **534**, **540**, **546**, **552**, **558**, **564**, **570**, **576**, **582**, **588**, **594**, **600**, **606**, **612**, **618**, **624**, **630**, **636**, **642**, **648**, **654**, **660**, **666**, **672**, **678**, **684**, **690**, **696**, **702**, **708**, **714**, **720**, **726**, **732**, **738**, **744**, **750**, **756**, **762**, **768**, **774**, **780**, **786**, **792**, **798**, **804**, **810**, **816**, **822**, **828**, **834**, **840**, **846**, **852**, **858**, **864**, **870**, **876**, **882**, **888**, **894**, **900**, **906**, **912**, **918**, **924**, **930**, **936**, **942**, **948**, **954**, **960**, **966**, **972**, **978**, **984**, **990**, **996** years would be sure to meet them.

Number theory

Prime numbers

Prime numbers in nature

- ▶ One example of the use of prime numbers in nature is as an evolutionary strategy used by **cicadas** of the genus **Magicicada**.
- ▶ These insects spend most of their lives as **grubs** underground.
- ▶ They only pupate and then emerge from their burrows after **13** or **17** years, at which point they fly about, breed, and then die after a few weeks at most.
- ▶ The logic for this is believed to be that the prime number intervals between emergences make it very difficult for predators to evolve that could specialize as predators on Magicicadas.
- ▶ If Magicicadas appeared at a non-prime number intervals, say every **12** years, then predators appearing every **2, 3, 4, 6, or 12** years would be sure to meet them.

Number theory

Prime numbers

Prime numbers in nature

- ▶ One example of the use of prime numbers in nature is as an evolutionary strategy used by **cicadas** of the genus **Magicicada**.
- ▶ These insects spend most of their lives as **grubs** underground.
- ▶ They only pupate and then emerge from their burrows after **13** or **17** years, at which point they fly about, breed, and then die after a few weeks at most.
- ▶ The logic for this is believed to be that the prime number intervals between emergences make it very difficult for predators to evolve that could specialize as predators on Magicicadas.
- ▶ If Magicicadas appeared at a non-prime number intervals, say every **12 years**, then predators appearing every **2, 3, 4, 6, or 12** years would be sure to meet them.

Number theory

Prime numbers

Prime numbers in nature

- ▶ One example of the use of prime numbers in nature is as an evolutionary strategy used by **cicadas** of the genus **Magicicada**.
- ▶ These insects spend most of their lives as **grubs** underground.
- ▶ They only pupate and then emerge from their burrows after **13** or **17** years, at which point they fly about, breed, and then die after a few weeks at most.
- ▶ The logic for this is believed to be that the prime number intervals between emergences make it very difficult for predators to evolve that could specialize as predators on Magicicadas.
- ▶ If Magicicadas appeared at a non-prime number intervals, say every **12 years**, then predators appearing every **2, 3, 4, 6**, or **12** years would be sure to meet them.

Number theory

Prime numbers

Prime numbers in nature

- ▶ One example of the use of prime numbers in nature is as an evolutionary strategy used by **cicadas** of the genus **Magicicada**.
- ▶ These insects spend most of their lives as **grubs** underground.
- ▶ They only pupate and then emerge from their burrows after **13** or **17** years, at which point they fly about, breed, and then die after a few weeks at most.
- ▶ The logic for this is believed to be that the prime number intervals between emergences make it very difficult for predators to evolve that could specialize as predators on Magicicadas.
- ▶ If Magicicadas appeared at a non-prime number intervals, say every **12 years**, then predators appearing every **2**, **3**, **4**, **6**, or **12** years would be sure to meet them.

Number theory

Prime numbers



Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\gcd(a, b)$
- ▶ We have $\gcd(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\text{gcd}(a, b)$
- ▶ We have $\text{gcd}(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\text{gcd}(a, b)$
- ▶ We have $\text{gcd}(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\gcd(a, b)$
- ▶ We have $\gcd(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\gcd(a, b)$
- ▶ We have $\gcd(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\gcd(a, b)$
- ▶ We have $\gcd(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

Find the GCD of 120 and 180.

Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\gcd(a, b)$
- ▶ We have $\gcd(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

- ▶ $\gcd(53, 71) = 1$ (*53 and 71 are primes*)
- ▶ $\gcd(18, 27) = 9$

Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\gcd(a, b)$
- ▶ We have $\gcd(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

- ▶ $\gcd(53, 71) = 1$ (*53 and 71 are primes*)
- ▶ $\gcd(18, 27) = 9$

Number theory

Greatest Common Divisor

Definition

- ▶ Let a and b be integers not both zero
- ▶ The greatest common divisor (GCD) of a and b is the largest positive integer which is a factor of both a and b
- ▶ The GCD of a and b is denoted $\gcd(a, b)$
- ▶ We have $\gcd(0, 0) = 0$
- ▶ The GCD of any two distinct primes is 1

Examples

- ▶ $\gcd(53, 71) = 1$ (*53 and 71 are primes*)
- ▶ $\gcd(18, 27) = 9$

Number theory

Euler's phi function

Introduction

- ▶ Consider all the integers in the interval $[1, 20]$
- ▶ The list of integers n where $1 \leq n \leq 20$ such that $\gcd(n, 20) = 1$ is
 $I = \{1, 3, 7, 9, 11, 13, 17, 19\}$
- ▶ So we have 8 integers in the closed interval $[1, 20]$ that are coprime to 20
- ▶ The Euler's phi function counts the number of integers $n < p$, such that $\gcd(n, p) = 1$

Definition

Number theory

Euler's phi function

Introduction

- ▶ Consider all the integers in the interval $[1, 20]$
- ▶ The list of integers n where $1 \leq n \leq 20$ such that $\gcd(n, 20) = 1$ is
 $I = \{1, 3, 7, 9, 11, 13, 17, 19\}$
- ▶ So we have 8 integers in the closed interval $[1, 20]$ that are coprime to 20
- ▶ The Euler's phi function counts the number of integers $a \in [1, n]$, such that
 $\gcd(a, n) = 1$

Definition

Number theory

Euler's phi function

Introduction

- ▶ Consider all the integers in the interval $[1, 20]$
- ▶ The list of integers n where $1 \leq n \leq 20$ such that $\gcd(n, 20) = 1$ is $I = \{1, 3, 7, 9, 11, 13, 17, 19\}$
- ▶ So we have 8 integers in the closed interval $[1, 20]$ that are coprime to 20
- ▶ The Euler's phi function counts the number of integers $a \in [1, n]$, such that $\gcd(a, n) = 1$

Definition

Number theory

Euler's phi function

Introduction

- ▶ Consider all the integers in the interval $[1, 20]$
- ▶ The list of integers n where $1 \leq n \leq 20$ such that $\gcd(n, 20) = 1$ is $I = \{1, 3, 7, 9, 11, 13, 17, 19\}$
- ▶ So we have 8 integers in the closed interval $[1, 20]$ that are coprime to 20
- ▶ The Euler's phi function counts the number of integers $a \in [1, n]$, such that $\gcd(a, n) = 1$

Definition

Number theory

Euler's phi function

Introduction

- ▶ Consider all the integers in the interval $[1, 20]$
- ▶ The list of integers n where $1 \leq n \leq 20$ such that $\gcd(n, 20) = 1$ is
 $I = \{1, 3, 7, 9, 11, 13, 17, 19\}$
- ▶ So we have 8 integers in the closed interval $[1, 20]$ that are coprime to 20
- ▶ The Euler's phi function counts the number of integers $a \in [1, n]$, such that
 $\gcd(a, n) = 1$

Definition

- ▶ The totient function $\phi(n)$, also called Euler's totient or Euler's phi function, is defined as the number of positive integers $\leq n$ that are relatively prime to n .

Number theory

Euler's phi function

Introduction

- ▶ Consider all the integers in the interval $[1, 20]$
- ▶ The list of integers n where $1 \leq n \leq 20$ such that $\gcd(n, 20) = 1$ is $I = \{1, 3, 7, 9, 11, 13, 17, 19\}$
- ▶ So we have 8 integers in the closed interval $[1, 20]$ that are coprime to 20
- ▶ The Euler's phi function counts the number of integers $a \in [1, n]$, such that $\gcd(a, n) = 1$

Definition

- ▶ The totient function $\phi(n)$, also called Euler's totient or Euler's phi function, is defined as the number of positive integers $\leq n$ that are relatively prime to n .

Number theory

Euler's phi function

Introduction

- ▶ Consider all the integers in the interval $[1, 20]$
- ▶ The list of integers n where $1 \leq n \leq 20$ such that $\gcd(n, 20) = 1$ is $I = \{1, 3, 7, 9, 11, 13, 17, 19\}$
- ▶ So we have 8 integers in the closed interval $[1, 20]$ that are coprime to 20
- ▶ The Euler's phi function counts the number of integers $a \in [1, n]$, such that $\gcd(a, n) = 1$

Definition

- ▶ The totient function $\phi(n)$, also called Euler's totient or Euler's phi function, is defined as the number of positive integers $\leq n$ that are relatively prime to n .

Number theory

Congruence

Practical case

Let us imagine: "Alice should encounter Bernard at 8:00 p.m.
Unfortunately it has 18 hours of delay. At what time will they meet?"



Résultat

• This is arithmetic modulo 24

The answer is:

8 + 18 = 26 → 26 mod 24 = 2

Number theory

Congruence

Practical case

Let us imagine: "Alice should encounter Bernard at 8:00 p.m.
Unfortunately it has 18 hours of delay. At what time will they meet?"



Résultat

- ▶ This is arithmetic modulo 24
- ▶ The answer is $(20 + 18) \text{ mod } 24 = 14$
- ▶ They will encounter next day at 2:00 p.m.
- ▶ In our case, we can say that 14 and 38 are equivalents modulo 24 which can be written $(20 + 18) \equiv 14 \pmod{24}$

Number theory

Congruence

Practical case

Let us imagine: "Alice should encounter Bernard at 8:00 p.m.
Unfortunately it has 18 hours of delay. At what time will they meet?"



Résultat

- ▶ This is arithmetic modulo 24
- ▶ The answer is $(20 + 18) \bmod 24 = 14$
- ▶ They will encounter next day at 2:00 p.m.
- ▶ In our case, we can say that 14 and 38 are equivalents modulo 24 which can be written $(20 + 18) \equiv 14 \pmod{24}$

Number theory

Congruence

Practical case

Let us imagine: "Alice should encounter Bernard at 8:00 p.m.
Unfortunately it has 18 hours of delay. At what time will they meet?"



Résultat

- ▶ This is arithmetic modulo 24
- ▶ The answer is $(20 + 18) \text{ mod } 24 = 14$
- ▶ They will encounter next day at 2:00 p.m.
- ▶ In our case, we can say that 14 and 38 are equivalents modulo 24 which can be written $(20 + 18) \equiv 14 \pmod{24}$

Number theory

Congruence

Practical case

Let us imagine: "Alice should encounter Bernard at 8:00 p.m.
Unfortunately it has 18 hours of delay. At what time will they meet?"



Résultat

- ▶ This is arithmetic modulo 24
- ▶ The answer is $(20 + 18) \text{ mod } 24 = 14$
- ▶ They will encounter next day at 2:00 p.m.
- ▶ In our case, we can say that 14 and 38 are equivalents modulo 24 which can be written $(20 + 18) \equiv 14 \pmod{24}$

Number theory

Congruence

Practical case

Let us imagine: "Alice should encounter Bernard at 8:00 p.m.
Unfortunately it has 18 hours of delay. At what time will they meet?"



Résultat

- ▶ This is arithmetic modulo 24
- ▶ The answer is $(20 + 18) \text{ mod } 24 = 14$
- ▶ They will encounter next day at 2:00 p.m.
- ▶ In our case, we can say that 14 and 38 are equivalents modulo 24 which can be written $(20 + 18) \equiv 14 \pmod{24}$

Number theory

Congruence

Definition

Congruence is an equivalence relation between two elements a and b such that $a \equiv b \pmod{n}$ with $a = b + kn, k \in \mathbb{Z}$



+



=

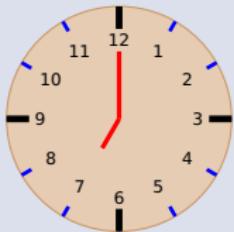


Number theory

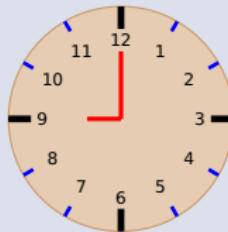
Congruence

Definition

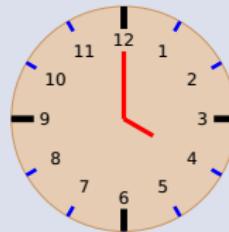
Congruence is an equivalence relation between two elements a and b such that $a \equiv b \pmod{n}$ with $a = b + kn, k \in \mathbb{Z}$



+



=



Number theory

Modular arithmetic

Definition

Modular arithmetic is the arithmetic of **congruences**.

Modular arithmetic can be handled mathematically by introducing a **congruence** relation on the integers that is compatible with the operations of the ring of integers: addition, subtraction, and multiplication.

Number theory

Modular arithmetic

Definition

Modular arithmetic is the arithmetic of **congruences**.

Modular arithmetic can be handled mathematically by introducing a **congruence** relation on the integers that is compatible with the operations of the ring of integers: addition, subtraction, and multiplication.

- ▶ Modular arithmetic modifies usual arithmetic by only using the numbers $\{0, 1, 2, \dots, n - 1\}$, $n \in \mathbb{Z}$
- ▶ n is a fixed natural number called modulus
- ▶ Calculating sums, differences and products is done as usual, but whenever a negative number or a number greater than $n - 1$ occurs, it gets replaced by the remainder after division by n
- ▶ For instance, for $n = 5$, the sum $-3 + 2$ is instead of -1 , since -1 divided by 5 has remainder 4
- ▶ This relation is denoted

Number theory

Modular arithmetic

Definition

Modular arithmetic is the arithmetic of **congruences**.

Modular arithmetic can be handled mathematically by introducing a **congruence** relation on the integers that is compatible with the operations of the ring of integers: addition, subtraction, and multiplication.

- ▶ Modular arithmetic modifies usual arithmetic by only using the numbers $\{0, 1, 2, \dots, n - 1\}, n \in \mathbb{Z}$
- ▶ n is a fixed natural number called modulus
- ▶ Calculating sums, differences and products is done as usual, but whenever a negative number or a number greater than $n - 1$ occurs, it gets replaced by the remainder after division by n
- ▶ For instance, for $n = 7$, the sum $3 + 5 = 1$ instead of 8, since 8 divided by 7 has remainder 1
- ▶ This relation is denoted

Number theory

Modular arithmetic

Definition

Modular arithmetic is the arithmetic of **congruences**.

Modular arithmetic can be handled mathematically by introducing a **congruence** relation on the integers that is compatible with the operations of the ring of integers: addition, subtraction, and multiplication.

- ▶ Modular arithmetic modifies usual arithmetic by only using the numbers $\{0, 1, 2, \dots, n - 1\}, n \in \mathbb{Z}$
- ▶ n is a fixed natural number called modulus
- ▶ Calculating sums, differences and products is done as usual, but whenever a negative number or a number greater than $n - 1$ occurs, it gets replaced by the remainder after division by n
- ▶ For instance, for $n = 7$, the sum $3 + 5 = 1$ instead of 8, since 8 divided by 7 has remainder 1
- ▶ This relation is denoted $(3 + 5) \equiv 1 \pmod{7}$

Number theory

Modular arithmetic

Definition

Modular arithmetic is the arithmetic of **congruences**.

Modular arithmetic can be handled mathematically by introducing a **congruence** relation on the integers that is compatible with the operations of the ring of integers: addition, subtraction, and multiplication.

- ▶ Modular arithmetic modifies usual arithmetic by only using the numbers $\{0, 1, 2, \dots, n - 1\}, n \in \mathbb{Z}$
- ▶ n is a fixed natural number called modulus
- ▶ Calculating sums, differences and products is done as usual, but whenever a negative number or a number greater than $n - 1$ occurs, it gets replaced by the remainder after division by n
- ▶ For instance, for $n = 7$, the sum $3 + 5 = 1$ instead of 8, since 8 divided by 7 has remainder 1
- ▶ This relation is denoted $(3 + 5) \equiv 1 \pmod{7}$

Number theory

Modular arithmetic

Definition

Modular arithmetic is the arithmetic of **congruences**.

Modular arithmetic can be handled mathematically by introducing a **congruence** relation on the integers that is compatible with the operations of the ring of integers: addition, subtraction, and multiplication.

- ▶ Modular arithmetic modifies usual arithmetic by only using the numbers $\{0, 1, 2, \dots, n - 1\}, n \in \mathbb{Z}$
- ▶ n is a fixed natural number called modulus
- ▶ Calculating sums, differences and products is done as usual, but whenever a negative number or a number greater than $n - 1$ occurs, it gets replaced by the remainder after division by n
- ▶ For instance, for $n = 7$, the sum $3 + 5 = 1$ instead of 8, since 8 divided by 7 has remainder 1
- ▶ This relation is denoted $(3 + 5) \equiv 1 \pmod{7}$

Number theory

Modular arithmetic

Definition

Modular arithmetic is the arithmetic of **congruences**.

Modular arithmetic can be handled mathematically by introducing a **congruence** relation on the integers that is compatible with the operations of the ring of integers: addition, subtraction, and multiplication.

- ▶ Modular arithmetic modifies usual arithmetic by only using the numbers $\{0, 1, 2, \dots, n - 1\}, n \in \mathbb{Z}$
- ▶ n is a fixed natural number called modulus
- ▶ Calculating sums, differences and products is done as usual, but whenever a negative number or a number greater than $n - 1$ occurs, it gets replaced by the remainder after division by n
- ▶ For instance, for $n = 7$, the sum $3 + 5 = 1$ instead of 8, since 8 divided by 7 has remainder 1
- ▶ This relation is denoted $(3 + 5) \equiv 1 \pmod{7}$

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

Given three numbers a , b and n , the modular exponentiation $a^b \pmod{n}$ is the remainder of the division of a^b by n .

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

- ▶ Modular exponentiation is a type of exponentiation performed over a modulus
- ▶ It is particularly useful in computer science especially in the field of cryptography

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

- ▶ Modular exponentiation is a type of exponentiation performed over a modulus
- ▶ It is particularly useful in computer science especially in the field of cryptography

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

- ▶ Modular exponentiation is a type of exponentiation performed over a modulus
 - ▶ It is particularly useful in computer science especially in the field of cryptography
-
- ▶ Calculation of $c = b^x \pmod{m}$
 - ▶ One-Way Function:

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

- ▶ Modular exponentiation is a type of exponentiation performed over a modulus
 - ▶ It is particularly useful in computer science especially in the field of cryptography
-
- ▶ Calculation of $c \equiv b^e \pmod{m}$
 - ▶ One-Way Function:

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

- ▶ Modular exponentiation is a type of exponentiation performed over a modulus
 - ▶ It is particularly useful in computer science especially in the field of cryptography
-
- ▶ Calculation of $c \equiv b^e \pmod{m}$
 - ▶ One-Way Function:
 - Calculation of $c \equiv b^e \pmod{m}$ is easy
 - Calculation of b from c and m is hard

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

- ▶ Modular exponentiation is a type of exponentiation performed over a modulus
 - ▶ It is particularly useful in computer science especially in the field of cryptography
-
- ▶ Calculation of $c \equiv b^e \pmod{m}$
 - ▶ One-Way Function:
 - ▶ Calculation of $c \equiv b^e \pmod{m}$ is easy
 - ▶ Calculation of discrete logarithm (recover c , e and m from b) is difficult

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

- ▶ Modular exponentiation is a type of exponentiation performed over a modulus
 - ▶ It is particularly useful in computer science especially in the field of cryptography
-
- ▶ Calculation of $c \equiv b^e \pmod{m}$
 - ▶ One-Way Function:
 - ▶ Calculation of $c \equiv b^e \pmod{m}$ is easy
 - ▶ Calculation of discrete logarithm (recover c , e and m from b) is difficult

Number theory

Modular arithmetic

Cryptography makes extensive use of **modular arithmetic** because it can restrict the sizes of intermediate and final results.

The modular exponentiation

- ▶ Modular exponentiation is a type of exponentiation performed over a modulus
 - ▶ It is particularly useful in computer science especially in the field of cryptography
-
- ▶ Calculation of $c \equiv b^e \pmod{m}$
 - ▶ One-Way Function:
 - ▶ Calculation of $c \equiv b^e \pmod{m}$ is easy
 - ▶ Calculation of discrete logarithm (recover c , e and m from b) is difficult

Number theory

Modular arithmetic

Tutorial: modular exponentiation

- ▶ To write properly an asymmetric cryptography program we need a **fast exponentiation algorithm**
-
- ① Search on the net what is modular exponentiation
 - ② Study the different algorithms of calculation
 - ③ Write a program in python which compute quickly $c = b^e \pmod{m}$
 - ④ Test your program with those variables:
 - ▶ $b = 10^{200}$
 - ▶ $e = 10^{100}$
 - ▶ $m = 10^{200} + 157$

Number theory

Modular arithmetic

Tutorial: a solution

We want to compute $c \equiv b^e \pmod{m}$

Right-to-left binary method

- ▶ exponent e is converted to binary: $e = \sum_{i=0}^{n-1} a_i 2^i$ with:
 - ▶ $a_i = 0$ or $a_i = 1$ ($0 \leq i < n - 1$)
 - ▶ Then the value c can be written as:
 - ▶ The solution c is therefore:

Number theory

Modular arithmetic

Tutorial: a solution

We want to compute $c \equiv b^e \pmod{m}$

Right-to-left binary method

- exponent e is converted to binary: $e = \sum_{i=0}^{n-1} a_i 2^i$ with:

- $n =$ length of e in bits
 - $a_i = 0$ or $a_i = 1$ ($0 \leq i < n - 1$)

- Then the value b^e can be written as: $b^e = b \left(\sum_{i=0}^{n-1} a_i 2^i \right) = \boxed{b} \left(b^2 \right)^{\frac{n-1}{2}}$

- The solution is therefore:

Number theory

Modular arithmetic

Tutorial: a solution

We want to compute $c \equiv b^e \pmod{m}$

Right-to-left binary method

- ▶ exponent e is converted to binary: $e = \sum_{i=0}^{n-1} a_i 2^i$ with:
 - ▶ $n =$ length of e in bits
 - ▶ $a_i = 0$ or $a_i = 1$ ($0 \leq i < n - 1$)
- ▶ Then the value b^e can be written as: $b^e = b^{\left(\sum_{i=0}^{n-1} a_i 2^i\right)} = \prod_{i=0}^{n-1} (b^{2^i})^{a_i}$
- ▶ the solution c is therefore: $c = \prod_{i=0}^{n-1} (b^{2^i})^{a_i} \pmod{m}$

Number theory

Modular arithmetic

Tutorial: a solution

We want to compute $c \equiv b^e \pmod{m}$

Right-to-left binary method

- ▶ exponent e is converted to binary: $e = \sum_{i=0}^{n-1} a_i 2^i$ with:
 - ▶ $n =$ length of e in bits
 - ▶ $a_i = 0$ or $a_i = 1$ ($0 \leq i < n - 1$)
- ▶ Then the value b^e can be written as: $b^e = b^{\left(\sum_{i=0}^{n-1} a_i 2^i\right)} = \prod_{i=0}^{n-1} (b^{2^i})^{a_i}$
- ▶ the solution c is therefore: $c \equiv \prod_{i=0}^{n-1} (b^{2^i})^{a_i} \pmod{m}$

Number theory

Modular arithmetic

Tutorial: a solution

We want to compute $c \equiv b^e \pmod{m}$

Right-to-left binary method

- ▶ exponent e is converted to binary: $e = \sum_{i=0}^{n-1} a_i 2^i$ with:
 - ▶ $n =$ length of e in bits
 - ▶ $a_i = 0$ or $a_i = 1$ ($0 \leq i < n - 1$)
- ▶ Then the value b^e can be written as: $b^e = b^{\left(\sum_{i=0}^{n-1} a_i 2^i\right)} = \prod_{i=0}^{n-1} (b^{2^i})^{a_i}$
- ▶ the solution c is therefore: $c \equiv \prod_{i=0}^{n-1} (b^{2^i})^{a_i} \pmod{m}$

Number theory

Modular arithmetic

Tutorial: a solution

We want to compute $c \equiv b^e \pmod{m}$

Right-to-left binary method

- ▶ exponent e is converted to binary: $e = \sum_{i=0}^{n-1} a_i 2^i$ with:
 - ▶ $n =$ length of e in bits
 - ▶ $a_i = 0$ or $a_i = 1$ ($0 \leq i < n - 1$)
- ▶ Then the value b^e can be written as: $b^e = b^{\left(\sum_{i=0}^{n-1} a_i 2^i\right)} = \prod_{i=0}^{n-1} (b^{2^i})^{a_i}$
- ▶ the solution c is therefore: $c \equiv \prod_{i=0}^{n-1} (b^{2^i})^{a_i} \pmod{m}$

Number theory

Modular arithmetic

Example

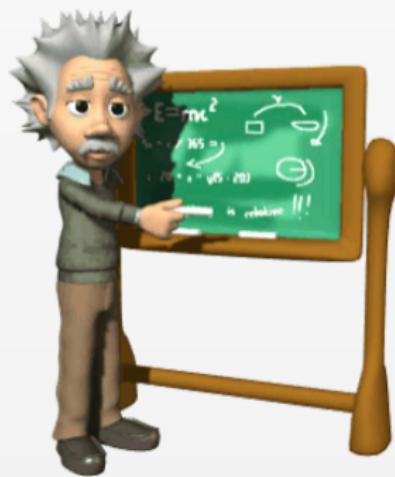
Number theory

Modular arithmetic

Example of code

```
def modexp(base, exponent, modulo):
    result = 1
    while exponent: # as the exponent is different from 0
        if (exponent%2): # if exponent is odd
            result = (result * base) % modulo # test
        exponent >>= 1 # we divide exponent by 2
        base = (base * base) % modulo
    return result % modulo
```

Boolean function



Boolean function

Definition

- ▶ A Boolean function is a function $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$, with $n > 1$
- ▶ A Boolean function is characterized by its truth table
- ▶ The arguments of Boolean functions are binary words of length n

Properties

- ▶ The support of a Boolean function $\text{supp}(f)$ is the set of elements of x such that $f(x) \neq 0$
- ▶ The weight of a Boolean function $\text{wt}(f)$ is the cardinal of its support

Boolean function

Definition

- ▶ A Boolean function is a function $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$, with $n > 1$
- ▶ A Boolean function is characterized by its truth table
- ▶ The arguments of Boolean functions are binary words of length n

Properties

- ▶ The support of a Boolean function $\text{supp}(f)$ is the set of elements of x such that $f(x) \neq 0$
- ▶ The weight of a Boolean function $\text{wt}(f)$ is the cardinal of its support

Boolean function

Definition

- ▶ A Boolean function is a function $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$, with $n > 1$
- ▶ A Boolean function is characterized by its truth table
- ▶ The arguments of Boolean functions are binary words of length n

Properties

- ▶ The support of a Boolean function $\text{supp}(f)$ is the set of elements of x such that $f(x) \neq 0$
- ▶ The weight of a Boolean function $\text{wt}(f)$ is the cardinal of its support

Boolean function

Definition

- ▶ A Boolean function is a function $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$, with $n > 1$
- ▶ A Boolean function is characterized by its truth table
- ▶ The arguments of Boolean functions are binary words of length n

Properties

- ▶ The support of a Boolean function $\text{supp}(f)$ is the set of elements of x such that $f(x) \neq 0$
- ▶ The weight of a Boolean function $\text{wt}(f)$ is the cardinal of its support

Boolean function

Definition

- ▶ A Boolean function is a function $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$, with $n > 1$
- ▶ A Boolean function is characterized by its truth table
- ▶ The arguments of Boolean functions are binary words of length n

Properties

- ▶ The support of a Boolean function $\text{supp}(f)$ is the set of elements of x such that $f(x) \neq 0$
- ▶ The weight of a Boolean function $\text{wt}(f)$ is the cardinal of its support

Boolean function

Boolean function representation

Algebraic Normal Form (ANF)

$$\begin{aligned} f(x_1, \dots, x_n) = & a_0 + a_1 x_1 + a_2 x_2 + \dots + a_n x_n \\ & + a_{1,2} x_1 x_2 + \dots + a_{n-1,n} x_{n-1} x_n \\ & + \dots + a_{1,2,\dots,n} x_1 x_2 \dots x_n \end{aligned}$$

Truth table

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	1

$$\begin{aligned} f(x_1, \dots, x_n) &= Q_f(x_1, \dots, x_n) \\ &= \sum_{(u_1, \dots, u_n) \in \mathbb{F}^n} a_u \prod_{i=1}^n x_i^{u_i} \end{aligned}$$

► Exists and unique

Boolean function

Boolean function – OR

Truth table

x_1	x_2	$x_1 \text{ OR } x_2$
0	0	0
0	1	1
1	0	1
1	1	1

Properties

- ▶ Notation $x_1 \vee x_2$
- ▶ $x_1 \vee x_2 = !(\neg x_1 \wedge \neg x_2)$

```
def logical_or(a, b):  
    return bool(a) | bool(b)
```

Boolean function

Boolean function – OR

Truth table

x_1	x_2	$x_1 \text{ OR } x_2$
0	0	0
0	1	1
1	0	1
1	1	1

Properties

- ▶ Notation $x_1 \vee x_2$
- ▶ $x_1 \vee x_2 = !(\neg x_1 \wedge \neg x_2)$

```
def logical_or(a, b):  
    return bool(a) | bool(b)
```

Boolean function

Boolean function – OR

Truth table

x_1	x_2	$x_1 \text{ OR } x_2$
0	0	0
0	1	1
1	0	1
1	1	1

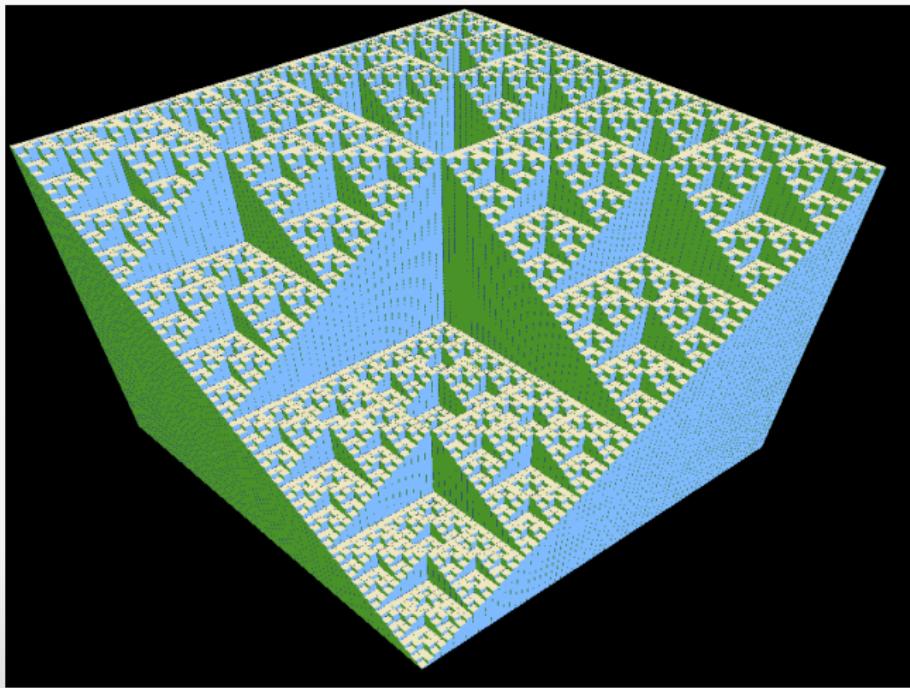
Properties

- ▶ Notation $x_1 \vee x_2$
- ▶ $x_1 \vee x_2 = \neg(\neg x_1 \wedge \neg x_2)$

```
def logical_or(a, b):  
    return bool(a) | bool(b)
```

Boolean function

Boolean function – OR



Boolean function

Boolean function – AND

Truth table

x_1	x_2	$x_1 \text{ AND } x_2$
0	0	0
0	1	0
1	0	0
1	1	1

Properties

- ▶ notation $x_1 \wedge x_2$
- ▶ $x_1 \wedge x_2 = !(\neg x_1 \vee \neg x_2)$

```
def logical_and(a, b):  
    return bool(a) & bool(b)
```

Boolean function

Boolean function – AND

Truth table

x_1	x_2	$x_1 \text{ AND } x_2$
0	0	0
0	1	0
1	0	0
1	1	1

Properties

- ▶ notation $x_1 \wedge x_2$
- ▶ $x_1 \wedge x_2 = \neg(\neg x_1 \vee \neg x_2)$

```
def logical_and(a, b):  
    return bool(a) & bool(b)
```

Boolean function

Boolean function – AND

Truth table

x_1	x_2	$x_1 \text{ AND } x_2$
0	0	0
0	1	0
1	0	0
1	1	1

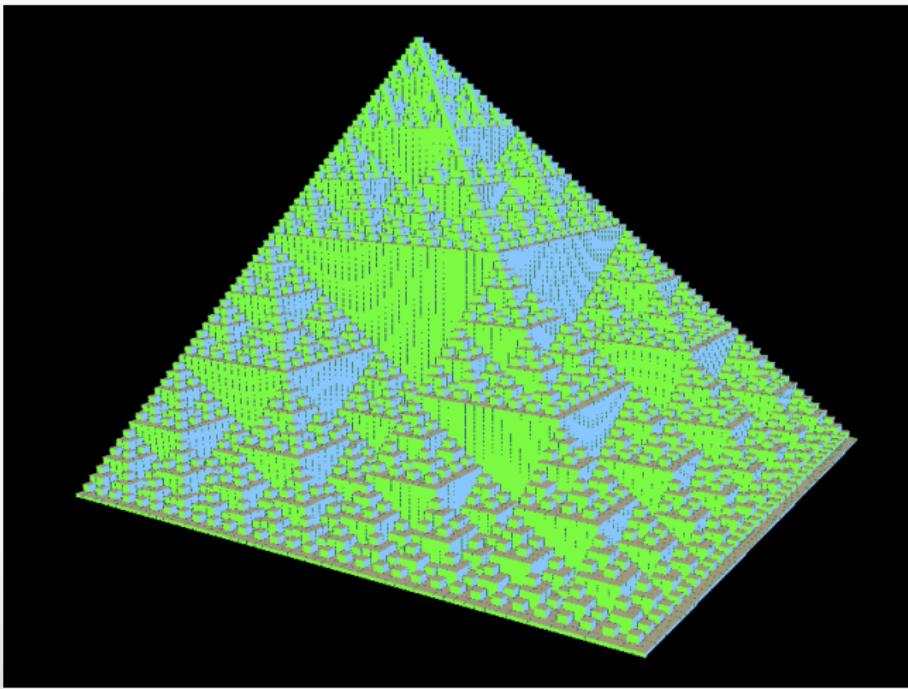
Properties

- ▶ notation $x_1 \wedge x_2$
- ▶ $x_1 \wedge x_2 = !(\neg x_1 \vee \neg x_2)$

```
def logical_and(a, b):  
    return bool(a) & bool(b)
```

Boolean function

Boolean function – AND



Boolean function

Boolean function – XOR

Truth table

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Properties

Commutative, associative

XOR swap algorithm

```
a = a ^ b  
b = a ^ b  
a = a ^ b
```

```
def logical_xor(a, b):  
    return bool(a) ^ bool(b)
```

Boolean function

Boolean function – XOR

Truth table

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

XOR swap algorithm

```
a = a ^ b
b = a ^ b
a = a ^ b
```

Properties

- ▶ notation $x_1 \text{ } \vee \text{ } x_2$
- ▶ $x_1 \vee x_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$
- ▶ Associativity: $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$
- ▶ Commutativity: $x_1 \vee x_2 = x_2 \vee x_1$
- ▶ Identity: $x_1 \vee 0 = x_1$
- ▶ Each element is its own inverse: $\neg x_1 = 0$

```
def logical_xor(a, b):
    return bool(a) ^ bool(b)
```

Boolean function

Boolean function – XOR

Truth table

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

XOR swap algorithm

```
a = a ^ b
b = a ^ b
a = a ^ b
```

Properties

- ▶ notation $x_1 \vee x_2$
- ▶ $x_1 \vee x_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$
- ▶ Associativity: $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$
- ▶ Commutativity: $x_1 \vee x_2 = x_2 \vee x_1$
- ▶ Identity: $x_1 \vee 0 = x_1$
- ▶ Each element is its own inverse: $x_1 \vee x_1 = 0$

```
def logical_xor(a, b):
    return bool(a) ^ bool(b)
```

Boolean function

Boolean function – XOR

Truth table

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

XOR swap algorithm

```
a = a ^ b  
b = a ^ b  
a = a ^ b
```

Properties

- ▶ notation $x_1 \vee x_2$
- ▶ $x_1 \vee x_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$
- ▶ Associativity: $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$
- ▶ Commutativity: $x_1 \vee x_2 = x_2 \vee x_1$
- ▶ Identity: $x_1 \vee 0 = x_1$
- ▶ Each element is its own inverse: $\neg x_1 = 0$

```
def logical_xor(a, b):  
    return bool(a) ^ bool(b)
```

Boolean function

Boolean function – XOR

Truth table

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

XOR swap algorithm

```
a = a ^ b  
b = a ^ b  
a = a ^ b
```

Properties

- ▶ notation $x_1 \text{ } \vee \text{ } x_2$
- ▶ $x_1 \vee x_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$
- ▶ Associativity: $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$
- ▶ Commutativity: $x_1 \vee x_2 = x_2 \vee x_1$
- ▶ Identity: $x_1 \vee 0 = x_1$
- ▶ Each element is its own inverse: $x_1 \vee x_1 = 0$

```
def logical_xor(a, b):  
    return bool(a) ^ bool(b)
```

Boolean function

Boolean function – XOR

Truth table

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

XOR swap algorithm

```
a = a ^ b
b = a ^ b
a = a ^ b
```

Properties

- ▶ notation $x_1 \text{ } \vee \text{ } x_2$
- ▶ $x_1 \vee x_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$
- ▶ Associativity: $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$
- ▶ Commutativity: $x_1 \vee x_2 = x_2 \vee x_1$
- ▶ Identity: $x_1 \vee 0 = x_1$
- ▶ Each element is its own inverse: $x_1 \vee x_1 = 0$

```
def logical_xor(a, b):
    return bool(a) ^ bool(b)
```

Boolean function

Boolean function – XOR

Truth table

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

XOR swap algorithm

```
a = a ^ b
b = a ^ b
a = a ^ b
```

Properties

- ▶ notation $x_1 \vee x_2$
- ▶ $x_1 \vee x_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$
- ▶ Associativity: $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$
- ▶ Commutativity: $x_1 \vee x_2 = x_2 \vee x_1$
- ▶ Identity: $x_1 \vee 0 = x_1$
- ▶ Each element is its own inverse: $x_1 \vee x_1 = 0$

```
def logical_xor(a, b):
    return bool(a) ^ bool(b)
```

Boolean function

Boolean function – XOR

Truth table

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

XOR swap algorithm

```
a = a ^ b
b = a ^ b
a = a ^ b
```

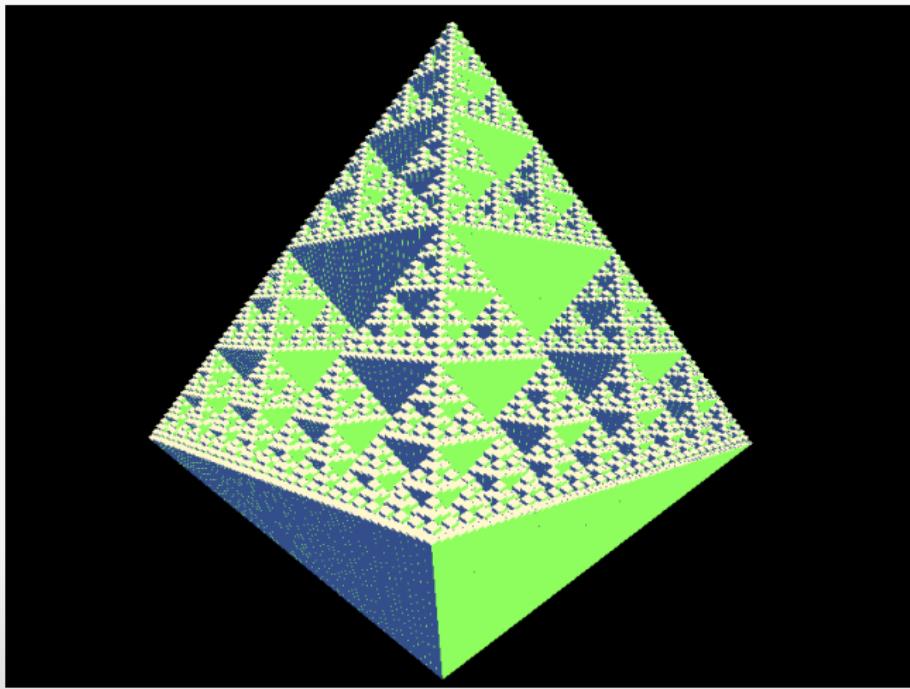
Properties

- ▶ notation $x_1 \vee x_2$
- ▶ $x_1 \vee x_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)$
- ▶ Associativity: $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$
- ▶ Commutativity: $x_1 \vee x_2 = x_2 \vee x_1$
- ▶ Identity: $x_1 \vee 0 = x_1$
- ▶ Each element is its own inverse: $x_1 \vee x_1 = 0$

```
def logical_xor(a, b):
    return bool(a) ^ bool(b)
```

Boolean function

Boolean function – XOR



Boolean function

Tutorial: XOR as cipher algorithm

- ① Write a function which takes as inputs:
 - ▶ a key
 - ▶ a file name to cipher
 - ▶ the name of the output file
- ② The cipher algorithm consists to do a cyclic XOR between the key and the data of the input file
- ③ Implement this function in C or Python and test it on some files

Boolean function

Tutorial: XOR as cipher algorithm – a solution

```
#include <stdio.h>

int main(int argc, char* argv[]) {
    FILE* fi;
    FILE* fo;
    char* key;
    unsigned int c;
    if ((key = argv[1])) {
        printf("Key: %s\n", key);
        if ((fi = fopen(argv[2], "r")) != NULL) {
            if ((fo = fopen(argv[3], "w")) != NULL) {
                while ((c=getc(fi)) != EOF) {
                    if (!*key) key = argv[1];
                    c ^= *(key++);
                    putc(c, fo);
                }
            }
            fclose(fo);
        }
        fclose(fi);
    }
    return(0);
}
```

Information Theory

2 Mathematical background

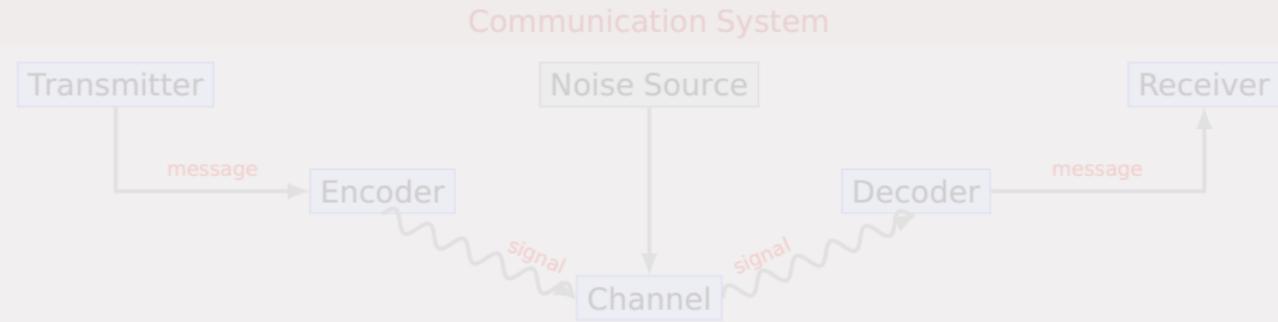
- Notation
- Mathematics for cryptography
 - Analysis
 - Abstract algebra
 - Number theory
 - Boolean function
- **Information Theory**
- Complexity theory



Information Theory

Information Theory

The **information theory** was developed by **Claude Elmwood Shannon** in his article “*A Mathematical Theory of Communications*” published in 1948.



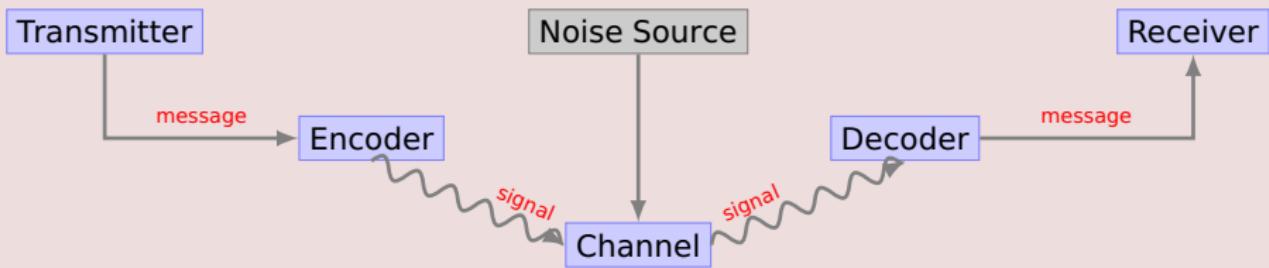
“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.”

Information Theory

Information Theory

The **information theory** was developed by **Claude Elmwood Shannon** in his article “*A Mathematical Theory of Communications*” published in 1948.

Communication System



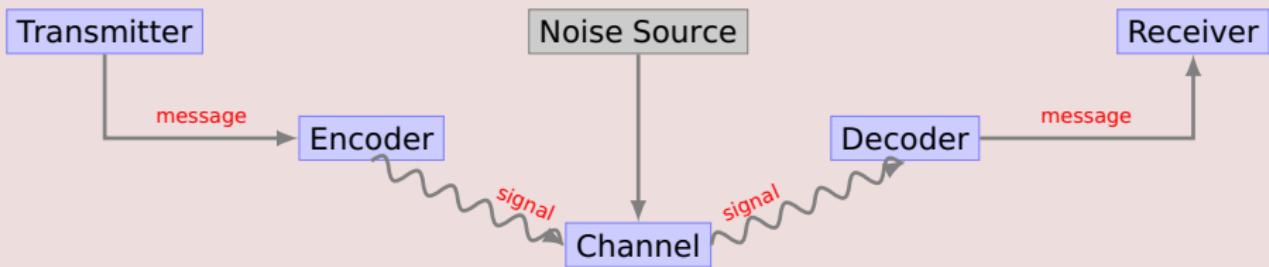
“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.”

Information Theory

Information Theory

The **information theory** was developed by **Claude Elmwood Shannon** in his article “*A Mathematical Theory of Communications*” published in 1948.

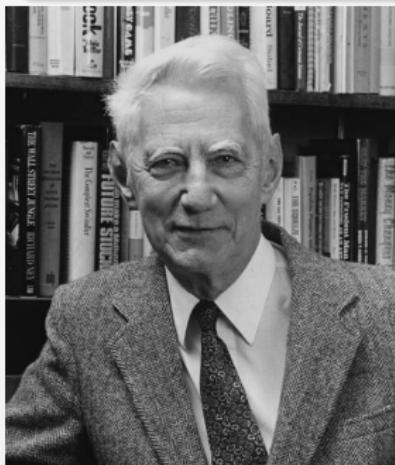
Communication System



*“The fundamental problem of communication is that of reproducing at one point either exactly or approximately a **message** selected at another point.”*

Information Theory

The information theory allows to **quantify** the information by determining the **information rate** used for the communication of a message through a noisy communication channel.



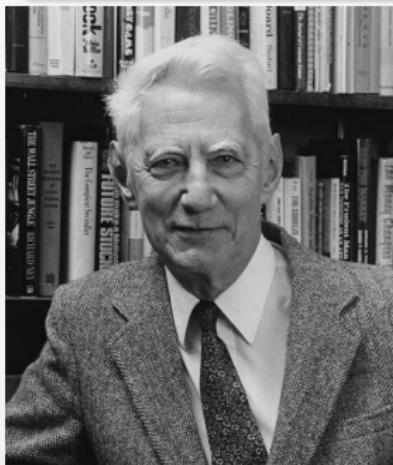
The information theory allows:

- ➊ to study the efficiency of communication: compression and error correcting codes
- ➋ to study the possibility of secret communication

Shannon generalizes the notion of **bit** as measure unit for information.

Information Theory

The information theory allows to **quantify** the information by determining the **information rate** used for the communication of a message through a noisy communication channel.



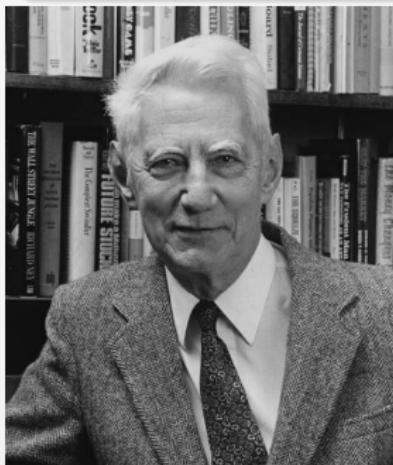
The information theory allows:

- ➊ to study the efficiency of communication: compression and error correcting codes
- ➋ to study the possibility of secret communication

Shannon generalizes the notion of **bit** as measure unit for information.

Information Theory

The information theory allows to **quantify** the information by determining the **information rate** used for the communication of a message through a noisy communication channel.



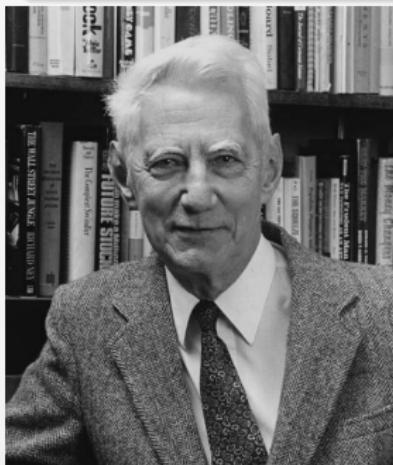
The information theory allows:

- ➊ to study the efficiency of communication: compression and error correcting codes
- ➋ to study the possibility of secret communication

Shannon generalizes the notion of **bit** as measure unit for information.

Information Theory

The information theory allows to **quantify** the information by determining the **information rate** used for the communication of a message through a noisy communication channel.



The information theory allows:

- ➊ to study the efficiency of communication: compression and error correcting codes
- ➋ to study the possibility of secret communication

Shannon generalizes the notion of **bit** as measure unit for information.

Information Theory

The entropy

Question

What is the amount of information needed to code the following data?

► men	► 1	► Sunday	► 000	► a	► 000001
► women	► 0	► Monday	► 001	► ...	► ...
		► Tuesday	► 010	► z	► 011010
		► Wednesday	► 011	► A	► 011011
		► Thursday	► 100	► ...	► ...
		► Friday	► 101	► z	► 110100
		► Saturday	► 110	► 0	► 110101
				► ...	► ...
				► 9	► 111110

Entropy of a message

The entropy of the message M is the minimum numbers of bits needed to code all possible meaning n of this message. The equation of the entropy is: $H(M) = \log_2 n$.

Information Theory

The entropy

Question

What is the amount of information needed to code the following data?

► men	► 1	► Sunday	► 000	► a	► 000001
► women	► 0	► Monday	► 001	► ...	► ...
		► Tuesday	► 010	► z	► 011010
		► Wednesday	► 011	► A	► 011011
		► Thursday	► 100	► ...	► ...
		► Friday	► 101	► z	► 110100
		► Saturday	► 110	► 0	► 110101
				► ...	► ...
				► 9	► 111110

Entropy of a message

The entropy of the message M is the minimum numbers of bits needed to code all possible meaning n of this message. The equation of the entropy is: $H(M) = \log_2 n$.

Information Theory

The entropy

Question

What is the amount of information needed to code the following data?

- ▶ men ▶ 1
- ▶ women ▶ 0

▶ Sunday	▶ 000	▶ a	▶ 000001
▶ Monday	▶ 001	▶ ...	▶ ...
▶ Tuesday	▶ 010	▶ z	▶ 011010
▶ Wednesday	▶ 011	▶ A	▶ 011011
▶ Thursday	▶ 100	▶ ...	▶ ...
▶ Friday	▶ 101	▶ z	▶ 110100
▶ Saturday	▶ 110	▶ 0	▶ 110101
		▶ ...	▶ ...
		▶ 9	▶ 111110

Entropy of a message

The entropy of the message M is the minimum numbers of bits needed to code all possible meaning n of this message. The equation of the entropy is: $H(M) = \log_2 n$.

Information Theory

The entropy

Question

What is the amount of information needed to code the following data?

► men	► 1	► Sunday	► 000	► a	► 000001
► women	► 0	► Monday	► 001	► ...	► ...
		► Tuesday	► 010	► z	► 011010
		► Wednesday	► 011	► A	► 011011
		► Thursday	► 100	► ...	► ...
		► Friday	► 101	► z	► 110100
		► Saturday	► 110	► 0	► 110101
				► ...	► ...
				► 9	► 111110

Entropy of a message

The entropy of the message M is the minimum numbers of bits needed to code all possible meaning n of this message. The equation of the entropy is: $H(M) = \log_2 n$.

Information Theory

The entropy

Question

What is the amount of information needed to code the following data?

► men	► 1	► Sunday	► 000	► a	► 000001
► women	► 0	► Monday	► 001	► ...	► ...
		► Tuesday	► 010	► z	► 011010
		► Wednesday	► 011	► A	► 011011
		► Thursday	► 100	► ...	► ...
		► Friday	► 101	► z	► 110100
		► Saturday	► 110	► 0	► 110101
				► ...	► ...
				► 9	► 111110

Entropy of a message

The **entropy** of the message M is the minimum numbers of bits needed to code all possible meaning n of this message. The equation of the entropy is: $H(M) = \log_2 n$.

Information Theory

The entropy

Question

What is the amount of information needed to code the following data?

► men	► 1	► Sunday	► 000	► a	► 000001
► women	► 0	► Monday	► 001	► ...	► ...
		► Tuesday	► 010	► z	► 011010
		► Wednesday	► 011	► A	► 011011
		► Thursday	► 100	► ...	► ...
		► Friday	► 101	► z	► 110100
		► Saturday	► 110	► 0	► 110101
				► ...	► ...
				► 9	► 111110

Entropy of a message

The **entropy** of the message M is the minimum numbers of bits needed to code all possible meaning n of this message. The equation of the entropy is: $H(M) = \log_2 n$.

Information Theory

The entropy

Question

What is the amount of information needed to code the following data?

► men	► 1	► Sunday	► 000	► a	► 000001
► women	► 0	► Monday	► 001	► ...	► ...
		► Tuesday	► 010	► z	► 011010
		► Wednesday	► 011	► A	► 011011
		► Thursday	► 100	► ...	► ...
		► Friday	► 101	► z	► 110100
		► Saturday	► 110	► 0	► 110101
				► ...	► ...
				► 9	► 111110

Entropy of a message

The **entropy** of the message M is the minimum numbers of bits needed to code all possible meaning n of this message. The equation of the entropy is: $H(M) = \log_2 n$.

Information Theory

The entropy

Question

What is the amount of information needed to code the following data?

► men	► 1	► Sunday	► 000	► a	► 000001
► women	► 0	► Monday	► 001	► ...	► ...
		► Tuesday	► 010	► z	► 011010
		► Wednesday	► 011	► A	► 011011
		► Thursday	► 100	► ...	► ...
		► Friday	► 101	► z	► 110100
		► Saturday	► 110	► 0	► 110101
				► ...	► ...
				► 9	► 111110

Entropy of a message

The **entropy** of the message M is the minimum numbers of bits needed to code all possible meaning n of this message. The equation of the entropy is: $H(M) = \log_2 n$.

Information Theory

The entropy

Entropy and Cryptography

With the **entropy** of a message, we can determine the number of necessary plain text bits to recover the plain text from a cipher text.

Example

If the ciphered text YHGTERSF has the meaning either dog or bird, then its entropy is 1.

Calm surface of water

No information
There is entropy

Water wave

There is **more** information
There is **less** entropy

Information Theory

The entropy

Entropy and Cryptography

With the **entropy** of a message, we can determine the number of necessary plain text bits to recover the plain text from a cipher text.

Example

If the ciphered text **YHGTERS**F has the meaning either **dog** or **bird**, then its entropy is **1**.
Indeed we need to discover only **1** bit to recover the plain text.

Calm surface of water

No information
There is entropy

Water wave

There is **more** information
There is **less** entropy

Information Theory

The entropy

Entropy and Cryptography

With the **entropy** of a message, we can determine the number of necessary plain text bits to recover the plain text from a cipher text.

Example

If the ciphered text **YHGTERS** has the meaning either **dog** or **bird**, then its entropy is **1**.
Indeed we need to discover only **1** bit to recover the plain text.

Calm surface of water

No information
There is entropy

Water wave

There is **more** information
There is **less** entropy

Information Theory

The entropy

Entropy and Cryptography

With the **entropy** of a message, we can determine the number of necessary plain text bits to recover the plain text from a cipher text.

Example

If the ciphered text **YHGTERS**F has the meaning either **dog** or **bird**, then its entropy is **1**. Indeed we need to discover only **1** bit to recover the plain text.

Calm surface of water

No information
There is entropy

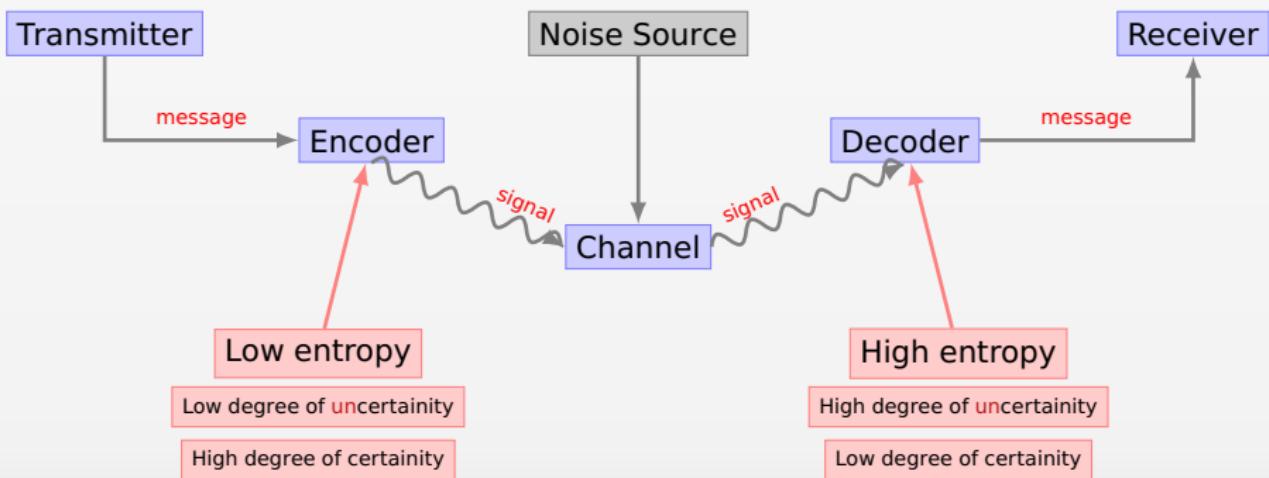
Water wave

There is **more** information
There is **less** entropy

Information Theory

The entropy

Understanding entropy



Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
- ▶ The **redundancy** of a language is defined by

($R - r$) / R which is the ratio between the absolute rate and the rate provided by the language.

Confusion and Diffusion

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1,3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english $R = 2,3$.
- ▶ The **redundancy** of a language is defined by
$$\text{Redundancy} = R - r$$
 and the smaller the redundancy the more information is lost.

Confusion and Diffusion

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1,3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
- ▶ The **redundancy** of a language is defined by $D = R - r$.

For example, English has a redundancy of about 0.6 bits per character, which corresponds to about 40% redundant information.

Confusion and Diffusion

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1, 3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4, 7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.

Confusion and Diffusion

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1, 3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4, 7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.

Confusion and Diffusion

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1,3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4,7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.

For english, $D = 4,7 - 1,3 = 3,4$. This means that each character provides 3,4 bits of redundant information.

Confusion and Diffusion

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1, 3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4, 7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.
For english, $D = 4, 7 - 1, 3 = 3, 4$. This means that each character provides 3, 4 bits of redundant information.

Confusion and Diffusion

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1, 3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4, 7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.
For english, $D = 4, 7 - 1, 3 = 3, 4$. This means that each character provides 3, 4 bits of redundant information.

Confusion and Diffusion

The confusion and diffusion principle is a well known principle in cryptography. It was introduced by Claude Shannon in his famous paper "A Mathematical Theory of Cryptography".

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1,3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4,7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.
For english, $D = 4,7 - 1,3 = 3,4$. This means that each character provides 3,4 bits of redundant information.

Confusion and Diffusion

- ▶ The **confusion** erases relationship between the message and the cryptogram.
The easiest way to achieve this is the **substitution**.
- ▶ The **diffusion** disperses the redundancy of the plaintext by spreading in the ciphertext.
The easiest way to achieve this is the **permutation**.

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1,3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4,7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.
For english, $D = 4,7 - 1,3 = 3,4$. This means that each character provides 3,4 bits of redundant information.

Confusion and Diffusion

- ▶ The **confusion** erases relationship between the message and the cryptogram.
The easiest way to achieve this is the **substitution**
- ▶ The **diffusion** disperses the redundancy of the plaintext by spreading in the ciphertext. The easiest way to achieve this is the **permutation**

Information Theory

Rate of language and redundancy

- Rate of language and redundancy
- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1,3$;
 - ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4,7$;
 - ▶ The **redundancy** of a language is defined by $D = R - r$.
For english, $D = 4,7 - 1,3 = 3,4$. This means that each character provides 3,4 bits of redundant information.

Confusion and Diffusion

- ▶ The **confusion** erases relationship between the message and the cryptogram.
The easiest way to achieve this is the **substitution**
- ▶ The **diffusion** disperses the redundancy of the plaintext by spreading in the ciphertext. The easiest way to achieve this is the **permutation**

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1,3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4,7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.
For english, $D = 4,7 - 1,3 = 3,4$. This means that each character provides 3,4 bits of redundant information.

Confusion and Diffusion

- ▶ The **confusion** erases relationship between the message and the cryptogram.
The easiest way to achieve this is the **substitution**
- ▶ The **diffusion** disperses the redundancy of the plaintext by spreading in the **ciphertext**. The easiest way to achieve this is the **permutation**

Information Theory

Rate of language and redundancy

Rate of language and redundancy

- ▶ The **rate of language**: $r = H(M)/N$ with N as the messages length).
For english $r = 1,3$;
- ▶ The **absolute rate** of a language is the maximum number of bits which can be encoded by each character: $R = \log_2 L$ where L is the number of characters of the language.
For english, $R = \log_2 26 = 4,7$;
- ▶ The **redundancy** of a language is defined by $D = R - r$.
For english, $D = 4,7 - 1,3 = 3,4$. This means that each character provides 3,4 bits of redundant information.

Confusion and Diffusion

- ▶ The **confusion** erases relationship between the message and the cryptogram.
The easiest way to achieve this is the **substitution**
- ▶ The **diffusion** disperses the redundancy of the plaintext by spreading in the ciphertext. The easiest way to achieve this is the **permutation**

Complexity theory

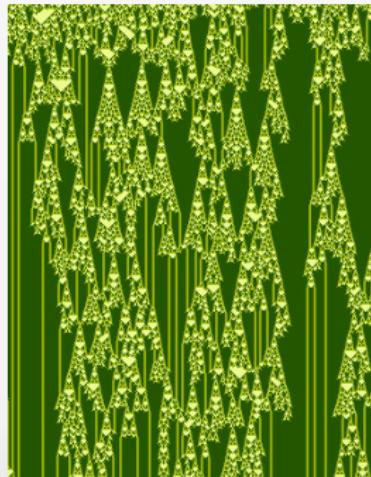
2 Mathematical background

- Notation
- Mathematics for cryptography
 - Analysis
 - Abstract algebra
 - Number theory
 - Boolean function
- Information Theory
- Complexity theory



Complexity theory

The main goal of **complexity theory** is to provide mechanisms for classifying computational problems according to the resources needed to solve them.



An **algorithm** is a well-defined computational procedure that takes a variable input and halts with an output.

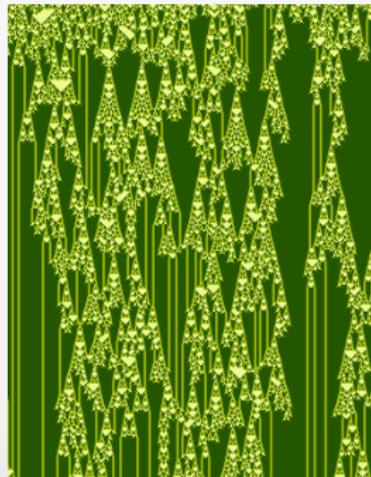
The **complexity** of an algorithm is determined by the computing power necessary to execute. It is measured by:

Time complexity

Space complexity

Complexity theory

The main goal of **complexity theory** is to provide mechanisms for classifying computational problems according to the resources needed to solve them.



An **algorithm** is a well-defined computational procedure that takes a variable input and halts with an output.

The **complexity** of an algorithm is determined by the computing power necessary to execute. It is measured by:

- ▶ T for time complexity
- ▶ S for memory space complexity
- ▶ n corresponding to the size of the input

Complexity theory

The main goal of **complexity theory** is to provide mechanisms for classifying computational problems according to the resources needed to solve them.



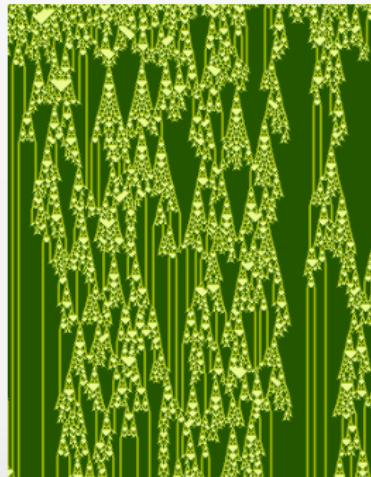
An **algorithm** is a well-defined computational procedure that takes a variable input and halts with an output.

The **complexity** of an algorithm is determined by the computing power necessary to execute. It is measured by:

- ▶ T for time complexity
- ▶ S for memory space complexity
- ▶ n corresponding to the size of the input

Complexity theory

The main goal of **complexity theory** is to provide mechanisms for classifying computational problems according to the resources needed to solve them.



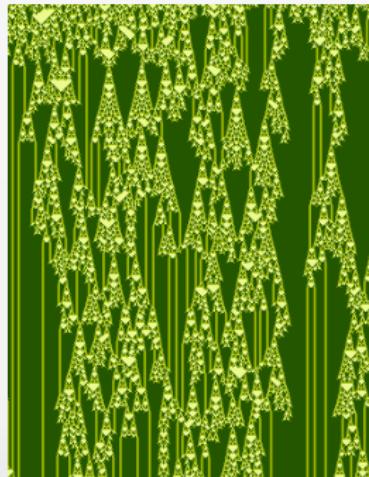
An **algorithm** is a well-defined computational procedure that takes a variable input and halts with an output.

The **complexity** of an algorithm is determined by the computing power necessary to execute. It is measured by:

- ▶ T for time complexity
- ▶ S for memory space complexity
- ▶ n corresponding to the size of the input

Complexity theory

The main goal of **complexity theory** is to provide mechanisms for classifying computational problems according to the resources needed to solve them.



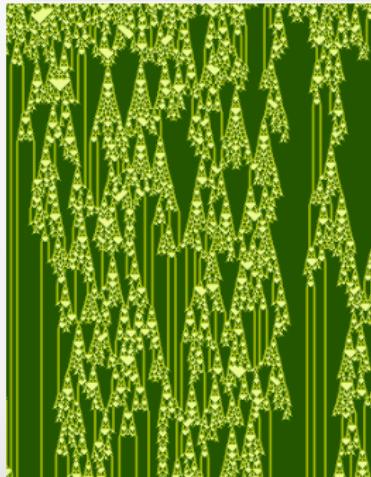
An **algorithm** is a well-defined computational procedure that takes a variable input and halts with an output.

The **complexity** of an algorithm is determined by the computing power necessary to execute. It is measured by:

- ▶ T for time complexity
- ▶ S for memory space complexity
- ▶ n corresponding to the size of the input

Complexity theory

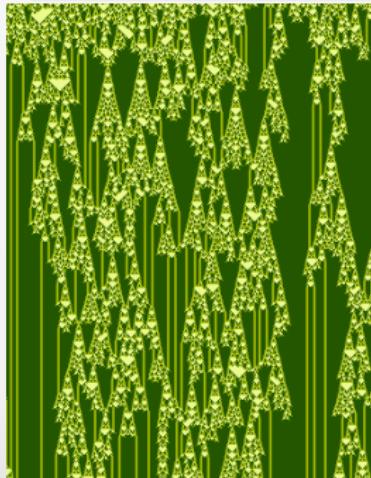
- ▶ If analyzing an algorithm, we find that the time required to solve a problem of size n is given by $T(n) = 4n^2 - 2n + 2$ then we can say that $T(n)$ is of order n^2 .
- ▶ Whether: $T(n) = \mathcal{O}(n^2)$



The big \mathcal{O} notation

Complexity theory

- ▶ If analyzing an algorithm, we find that the time required to solve a problem of size n is given by $T(n) = 4n^2 - 2n + 2$ then we can say that $T(n)$ is of order n^2 .
- ▶ Whether: $T(n) = \mathcal{O}(n^2)$

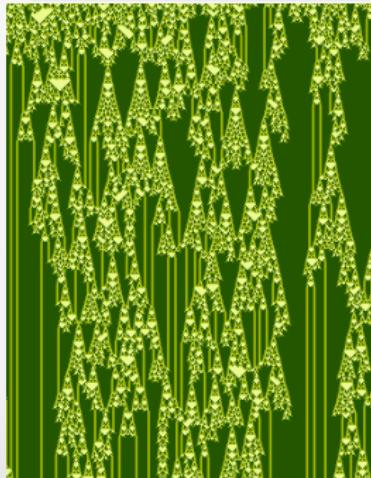


The big \mathcal{O} notation

Big O notation is a way to describe the performance of an algorithm. It tells us how the running time or space requirements grow as the input size increases.

Complexity theory

- ▶ If analyzing an algorithm, we find that the time required to solve a problem of size n is given by $T(n) = 4n^2 - 2n + 2$ then we can say that $T(n)$ is of order n^2 .
- ▶ Whether: $T(n) = \mathcal{O}(n^2)$

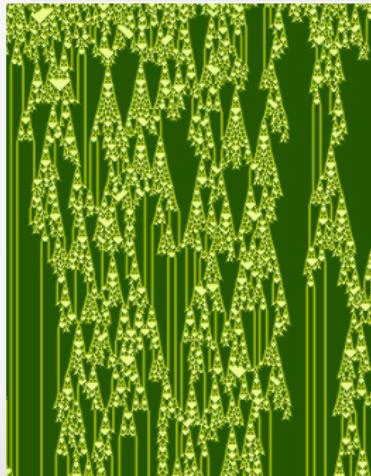


The big \mathcal{O} notation

It allows to express the magnitude of the computational complexity

Complexity theory

- ▶ If analyzing an algorithm, we find that the time required to solve a problem of size n is given by $T(n) = 4n^2 - 2n + 2$ then we can say that $T(n)$ is of order n^2 .
- ▶ Whether: $T(n) = \mathcal{O}(n^2)$

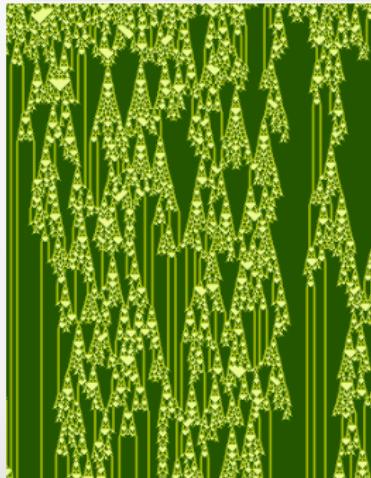


The big \mathcal{O} notation

- ▶ $\mathcal{O}(n)$ defines the magnitude of the computational complexity
- ▶ This notation allows us to see how the needs of time and space change with the size of the inputs

Complexity theory

- ▶ If analyzing an algorithm, we find that the time required to solve a problem of size n is given by $T(n) = 4n^2 - 2n + 2$ then we can say that $T(n)$ is of order n^2 .
- ▶ Whether: $T(n) = \mathcal{O}(n^2)$

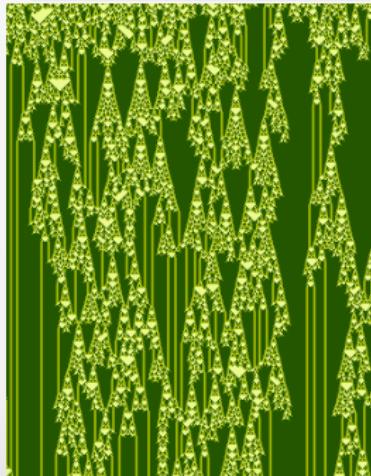


The big \mathcal{O} notation

- ▶ $\mathcal{O}(n)$ defines the magnitude of the computational complexity
- ▶ This notation allows us to see how the needs of time and space change with the size of the inputs

Complexity theory

- ▶ If analyzing an algorithm, we find that the time required to solve a problem of size n is given by $T(n) = 4n^2 - 2n + 2$ then we can say that $T(n)$ is of order n^2 .
- ▶ Whether: $T(n) = \mathcal{O}(n^2)$



The big \mathcal{O} notation

- ▶ $\mathcal{O}(n)$ defines the magnitude of the computational complexity
- ▶ This notation allows us to see how the needs of time and space change with the size of the inputs

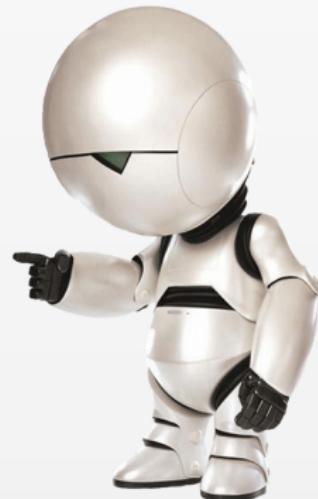
Complexity theory

Computation time for different classes of algorithms

Class	Complexity	Number of operations for $n = 10^6$	Time for 10^6 operations by second
Constant	$\mathcal{O}(1)$	1	$1\mu s$
Linear	$\mathcal{O}(n)$	10^6	1s
Quadratic	$\mathcal{O}(n^2)$	10^{12}	$11,6j$
Cubic	$\mathcal{O}(n^3)$	10^{18}	32000 années
Exponential	$\mathcal{O}(2^n)$	10^{301030}	10^{301006} times age of the universe

History

- 1** Introduction
- 2** Mathematical background
- 3** History
- 4** In practice...



felix the cat



- ▶ length = 200×200 pixels whether a square matrix of 200
- ▶ message length **40 000 characters**
- ▶ Picture in grayscale
- ▶ Alphabet = [0..255] classical alphabet = [0..255]
- ▶ Operations modulo 255 (numbers modulo 256)

felix the cat



- ▶ length = 200×200 pixels whether a square matrix of 200
- ▶ message length 40 000 characters
- ▶ Picture in grayscale
- ▶ Alphabet = [0..255] Classical alphabet = [0..25]
- ▶ Operations modulo 255 (modulo 20)

felix the cat



- ▶ length = 200×200 pixels whether a square matrix of 200
- ▶ message length **40 000 characters**
- ▶ Picture in grayscale
- ▶ Alphabet = [0..255] Classical alphabet = [0..25]
- ▶ Operations modulo 256 Operations modulo 26

felix the cat



- ▶ length = 200×200 pixels whether a square matrix of 200
- ▶ message length **40 000 characters**
- ▶ Picture in grayscale
- ▶ Alphabet = [0..255] Classical alphabet = [0..25]
- ▶ Operations modulo 256 Oprations modulo 26

felix the cat



- ▶ length = 200×200 pixels whether a square matrix of 200
- ▶ message length **40 000 characters**
- ▶ Picture in grayscale
- ▶ Alphabet = [0..255] Classical alphabet = [0..25]
- ▶ Operations modulo 256 Oprations modulo 26

felix the cat



- ▶ length = 200×200 pixels whether a square matrix of 200
- ▶ message length **40 000 characters**
- ▶ Picture in grayscale
- ▶ Alphabet = [0..255] Classical alphabet = [0..25]
- ▶ Operations modulo 256 Oprations modulo 26

First period – craftsmanship age: from the origins to WW I

3 History

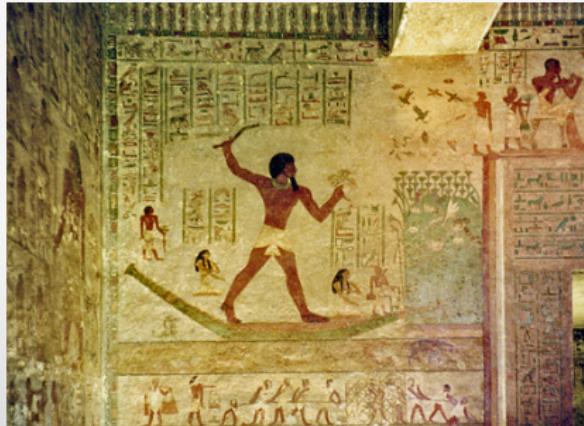
- First period – craftsmanship age: from the origins to WW I
- Second period – technical age: from WWI to Shannon's advent
- Third period – modern age: from the end of WW II to our days



Craftsmanship age: from the origins to WW I

First use of cryptography

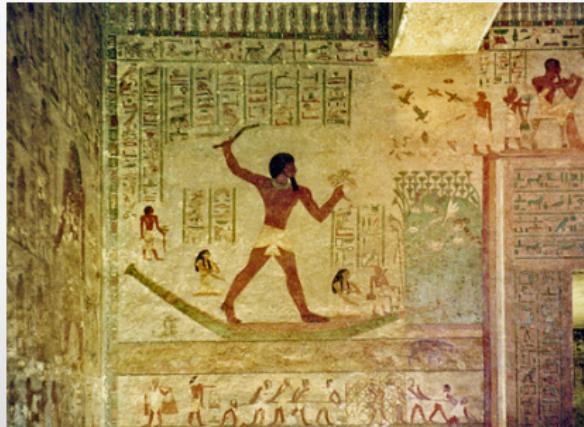
- ▶ The **oldest** known text to contain elements of cryptography occurred **2000 BC**
- ▶ It appears in the Egyptian town of **Menet Khufu** on the tomb of the nobleman **Khnumhotep II**
- ▶ The hieroglyphic inscriptions on the tomb were written with a number of unusual symbols to confuse the meaning of the inscriptions
- ▶ It was a system of



Craftsmanship age: from the origins to WW I

First use of cryptography

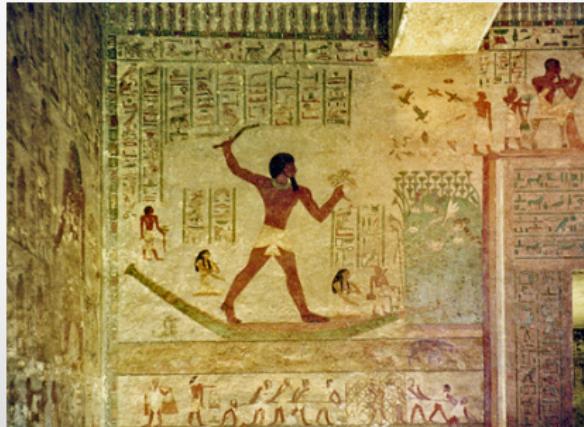
- ▶ The **oldest** known text to contain elements of cryptography occurred **2000 BC**
- ▶ It appears in the Egyptian town of **Menet Khufu** on the tomb of the nobleman **Khnumhotep II**
- ▶ The hieroglyphic inscriptions on the tomb were written with a number of unusual symbols to confuse the meaning of the inscriptions
- ▶ It was a system of partial substitution



Craftsmanship age: from the origins to WW I

First use of cryptography

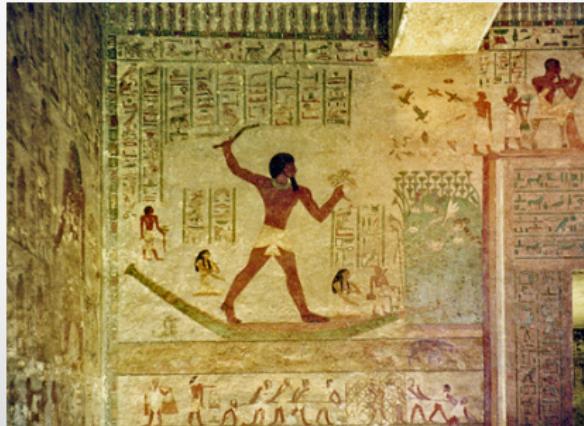
- ▶ The **oldest** known text to contain elements of cryptography occurred **2000 BC**
- ▶ It appears in the Egyptian town of **Menet Khufu** on the tomb of the nobleman **Khnumhotep II**
- ▶ The hieroglyphic inscriptions on the tomb were written with a number of unusual symbols to confuse the meaning of the inscriptions
- ▶ It was a system of **partial substitution**



Craftsmanship age: from the origins to WW I

First use of cryptography

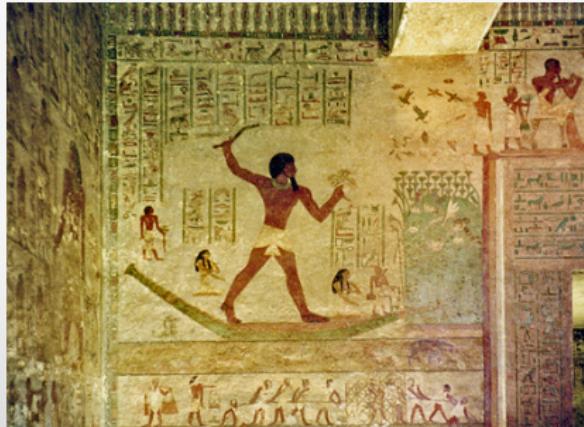
- ▶ The **oldest** known text to contain elements of cryptography occurred **2000 BC**
- ▶ It appears in the Egyptian town of **Menet Khufu** on the tomb of the nobleman **Khnumhotep II**
- ▶ The hieroglyphic inscriptions on the tomb were written with a number of unusual symbols to confuse the meaning of the inscriptions
- ▶ It was a system of **partial substitution**



Craftsmanship age: from the origins to WW I

First use of cryptography

- ▶ The **oldest** known text to contain elements of cryptography occurred **2000 BC**
- ▶ It appears in the Egyptian town of **Menet Khufu** on the tomb of the nobleman **Khnumhotep II**
- ▶ The hieroglyphic inscriptions on the tomb were written with a number of unusual symbols to confuse the meaning of the inscriptions
- ▶ It was a system of **partial substitution**



Craftsmanship age: from the origins to WW I

The Caesar cipher

The cipher of Caesar

Text is ciphered by shifting of 3 ranks the letters of the message.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: YHQL YLGL YLFL

Principles

Craftsmanship age: from the origins to WW I

The Caesar cipher

The cipher of Caesar

Text is ciphered by shifting of 3 ranks the letters of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **YHQL YLGL YLFL**

Principles

Caesar cipher is a monoalphabetic substitution cipher.

It consists in shifting the letters of the alphabet by a fixed number.

For example, with a shift of 3:

Message: **VENI VIDI VICI**

Cryptogram: **YHQL YLGL YLFL**

With a shift of 13:

Message: **VENI VIDI VICI**

Cryptogram: **XWQD XWQD XWQD**

Craftsmanship age: from the origins to WW I

The Caesar cipher

The cipher of Caesar

Text is ciphered by shifting of 3 ranks the letters of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **YHQL YLGL YLFL**

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C(x) = x + 3 \text{ mod } 26$

Craftsmanship age: from the origins to WW I

The Caesar cipher

The cipher of Caesar

Text is ciphered by shifting of 3 ranks the letters of the message.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: YHQL YLGL YLFL

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_k(x) = x + k \bmod 26$
- ▶ Deciphering: $D_k(y) = y - k \bmod 26$
- ▶ For Caesar's cipher, $k = 3$

Craftsmanship age: from the origins to WW I

The Caesar cipher

The cipher of Caesar

Text is ciphered by shifting of 3 ranks the letters of the message.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: YHQL YLGL YLFL

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_k(x) = x + k \bmod 26$
- ▶ Deciphering: $D_k(y) = y - k \bmod 26$
- ▶ For Caesar's cipher, $k = 3$

Craftsmanship age: from the origins to WW I

The Caesar cipher

The cipher of Caesar

Text is ciphered by shifting of 3 ranks the letters of the message.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: YHQL YLGL YLFL

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_k(x) = x + k \bmod 26$
- ▶ Deciphering: $D_k(y) = y - k \bmod 26$
- ▶ For Caesar's cipher, $k = 3$

Craftsmanship age: from the origins to WW I

The Caesar cipher

The cipher of Caesar

Text is ciphered by shifting of 3 ranks the letters of the message.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: YHQL YLGL YLFL

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_k(x) = x + k \bmod 26$
- ▶ Deciphering: $D_k(y) = y - k \bmod 26$
- ▶ For Caesar's cipher, $k = 3$

Craftsmanship age: from the origins to WW I

The Caesar cipher

The cipher of Caesar

Text is ciphered by shifting of 3 ranks the letters of the message.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: YHQL YLGL YLFL

Principles

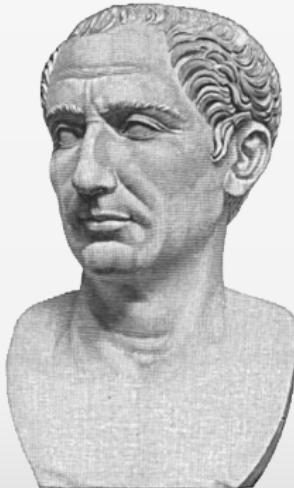
- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_k(x) = x + k \bmod 26$
- ▶ Deciphering: $D_k(y) = y - k \bmod 26$
- ▶ For Caesar's cipher, $k = 3$

Craftsmanship age: from the origins to WW I

The Caesar cipher

Cons

The Caesar cipher does not change the **entropy** of the message.
His cryptanalysis is easier if the ciphertext is long enough.



Craftsmanship age: from the origins to WW I

The Caesar cipher

Encryption by mono-alphabetic substitution: visualization



Shift of 32
Shift of 64

Shift of 96
Shift of 128

Shift of 160
Shift of 192

Shift of 224
Shift of 256

Craftsmanship age: from the origins to WW I

The Caesar cipher

Encryption by mono-alphabetic substitution: visualization



Shift of 32



Shift of 64

Shift of 96
Shift of 128

Shift of 160
Shift of 192

Shift of 224
Shift of 256

Craftsmanship age: from the origins to WW I

The Caesar cipher

Encryption by mono-alphabetic substitution: visualization



Shift of 32



Shift of 64



Shift of 96
Shift of 128

Shift of 160

Shift of 192

Shift of 224

Shift of 256

Craftsmanship age: from the origins to WW I

The Caesar cipher

Encryption by mono-alphabetic substitution: visualization



Shift of 32



Shift of 96



Shift of 64



Shift of 128

Shift of 160

Shift of 192

Shift of 224

Shift of 256

Craftsmanship age: from the origins to WW I

The Caesar cipher

Encryption by mono-alphabetic substitution: visualization



Shift of 32



Shift of 96



Shift of 224
Shift of 256



Shift of 64



Shift of 128

Craftsmanship age: from the origins to WW I

The Caesar cipher

Encryption by mono-alphabetic substitution: visualization



Shift of 32



Shift of 96



Shift of 160

Shift of 224

Shift of 256



Shift of 64



Shift of 128



Shift of 192

Craftsmanship age: from the origins to WW I

The Caesar cipher

Encryption by mono-alphabetic substitution: visualization



Shift of 32



Shift of 96



Shift of 160



Shift of 224
Shift of 256



Shift of 64



Shift of 128



Shift of 192

Craftsmanship age: from the origins to WW I

The Caesar cipher

Encryption by mono-alphabetic substitution: visualization



Shift of 32



Shift of 96



Shift of 160



Shift of 224



Shift of 64



Shift of 128



Shift of 192



Shift of 256

Craftsmanship age: from the origins to WW I

The Caesar cipher

Practice with Sage: Mono-alphabetic Substitution

```
# plaintext/ciphertext alphabet
A = AlphabeticStrings()
S = SubstitutionCryptosystem(A); S
P = "Substitute this with something else better."; P
K = A([ (i+3)%26 for i in range(26) ]); K

# encoding
msg = A.encoding(P); msg

# ciphering
C = S.enciphering(K, msg); C

# deciphering
DC = S.deciphering(K, C); DC

# control result
msg == DC
```

Craftsmanship age: from the origins to WW I

The affine cipher

The affine cipher

The plaintext is ciphered by applying to each character an affine function: $y = ax + b$

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: RSTE REPE REME

Principles

Craftsmanship age: from the origins to WW I

The affine cipher

The affine cipher

The plaintext is ciphered by applying to each character an affine function: $y = ax + b$

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: RSTE REPE REME

Principles

Encryption: $y = ax + b \pmod{26}$

Decryption: $x = (y - b) \cdot a^{-1} \pmod{26}$

Condition: $a \neq 0 \pmod{26}$

Number of keys: $\phi(26) = 20$

Number of ciphertexts: 26

Number of plaintexts: 26

Number of keys: $\phi(26) = 20$

Craftsmanship age: from the origins to WW I

The affine cipher

The affine cipher

The plaintext is ciphered by applying to each character an affine function: $y = ax + b$

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **RSTE REPE REME**

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $Crypt(x) = (ax + b) \bmod 26$

Craftsmanship age: from the origins to WW I

The affine cipher

The affine cipher

The plaintext is ciphered by applying to each character an affine function: $y = ax + b$

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: RSTE REPE REME

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_{(a,b)}(x) = (ax + b) \bmod 26$
- ▶ Deciphering: $D_{(a,b)}(y) = a^{-1}(y - b) \bmod 26$
- ▶ For our example, $a = 3$, $b = 6$

Craftsmanship age: from the origins to WW I

The affine cipher

The affine cipher

The plaintext is ciphered by applying to each character an affine function: $y = ax + b$

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: RSTE REPE REME

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_{(a,b)}(x) = (ax + b) \bmod 26$
- ▶ Deciphering: $D_{(a,b)}(y) = a^{-1}(y - b) \bmod 26$
- ▶ For our example, $a = 3$, $b = 6$

Craftsmanship age: from the origins to WW I

The affine cipher

The affine cipher

The plaintext is ciphered by applying to each character an affine function: $y = ax + b$

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: RSTE REPE REME

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_{(a,b)}(x) = (ax + b) \text{ mod } 26$
- ▶ Deciphering: $D_{(a,b)}(y) = a^{-1}(y - b) \text{ mod } 26$
- ▶ For our example, $a = 3$, $b = 6$

Craftsmanship age: from the origins to WW I

The affine cipher

The affine cipher

The plaintext is ciphered by applying to each character an affine function: $y = ax + b$

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: RSTE REPE REME

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_{(a,b)}(x) = (ax + b) \bmod 26$
- ▶ Deciphering: $D_{(a,b)}(y) = a^{-1}(y - b) \bmod 26$
- ▶ For our example, $a = 3$, $b = 6$

Craftsmanship age: from the origins to WW I

The affine cipher

The affine cipher

The plaintext is ciphered by applying to each character an affine function: $y = ax + b$

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: RSTE REPE REME

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ Ciphering: $C_{(a,b)}(x) = (ax + b) \bmod 26$
- ▶ Deciphering: $D_{(a,b)}(y) = a^{-1}(y - b) \bmod 26$
- ▶ For our example, $a = 3$, $b = 6$

Craftsmanship age: from the origins to WW I

The affine cipher

Encryption by mono-alphabetic substitution: visualization



$a=1, b=0$
 $a=1, b=40$

$a=1, b=100$
 $a=40, b=0$

$a=40, b=40$
 $a=40, b=100$

$a=100, b=0$
 $a=100, b=40$

Craftsmanship age: from the origins to WW I

The affine cipher

Encryption by mono-alphabetic substitution: visualization



$a=1, b=0$



$a=1, b=40$

$a=1, b=100$
 $a=40, b=0$

$a=40, b=40$
 $a=40, b=100$

$a=100, b=0$
 $a=100, b=40$

Craftsmanship age: from the origins to WW I

The affine cipher

Encryption by mono-alphabetic substitution: visualization



$a=1, b=0$



$a=1, b=40$



$a=1, b=100$
 $a=40, b=0$

$a=40, b=40$
 $a=40, b=100$

$a=100, b=0$
 $a=100, b=40$

Craftsmanship age: from the origins to WW I

The affine cipher

Encryption by mono-alphabetic substitution: visualization



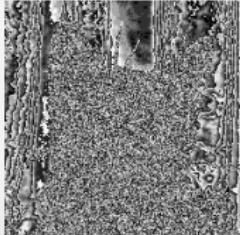
$a=1, b=0$



$a=1, b=100$



$a=1, b=40$



$a=40, b=0$

$a=40, b=40$

$a=40, b=100$

$a=100, b=0$

$a=100, b=40$

Craftsmanship age: from the origins to WW I

The affine cipher

Encryption by mono-alphabetic substitution: visualization



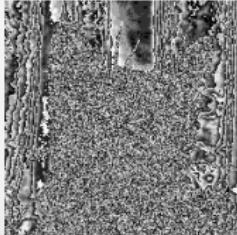
$a=1, b=0$



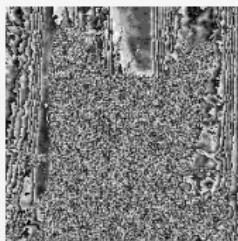
$a=1, b=100$



$a=1, b=40$



$a=40, b=0$



$a=40, b=40$

$a=40, b=100$

$a=100, b=0$

$a=100, b=40$

Craftsmanship age: from the origins to WW I

The affine cipher

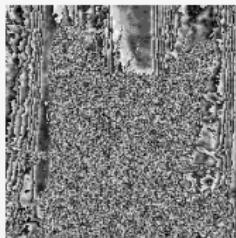
Encryption by mono-alphabetic substitution: visualization



$a=1, b=0$



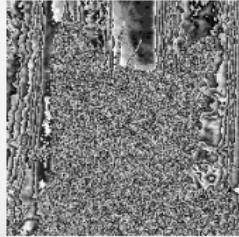
$a=1, b=100$



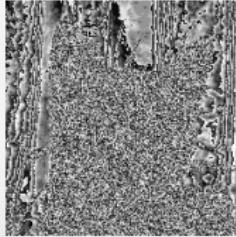
$a=40, b=40$



$a=1, b=40$



$a=40, b=0$



$a=40, b=100$

$a=100, b=0$
 $a=100, b=40$

Craftsmanship age: from the origins to WW I

The affine cipher

Encryption by mono-alphabetic substitution: visualization



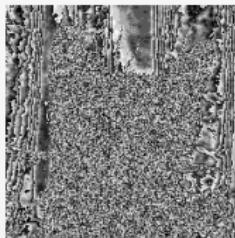
$a=1, b=0$



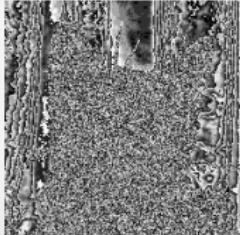
$a=1, b=100$



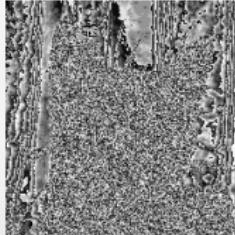
$a=1, b=40$



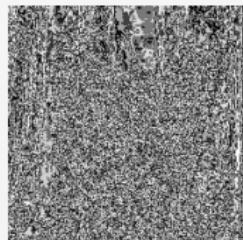
$a=40, b=40$



$a=40, b=0$



$a=40, b=100$



$a=100, b=0$
 $a=100, b=40$

Craftsmanship age: from the origins to WW I

The affine cipher

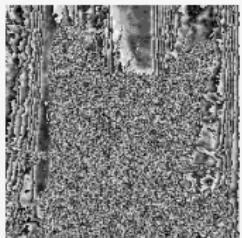
Encryption by mono-alphabetic substitution: visualization



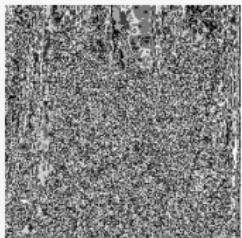
$a=1, b=0$



$a=1, b=100$



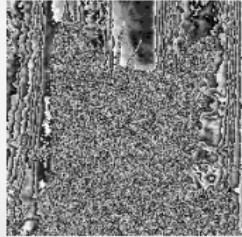
$a=40, b=40$



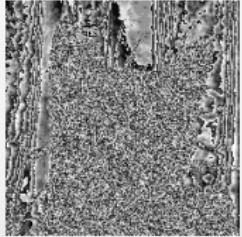
$a=100, b=0$



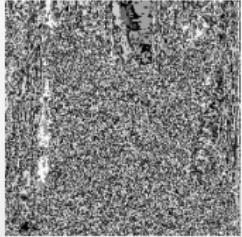
$a=1, b=40$



$a=40, b=0$



$a=40, b=100$



$a=100, b=40$

Craftsmanship age: from the origins to WW I

The affine cipher

Practice with Sage: **Affine cipher**

```
# create an affine cipher
affineCipher = AffineCryptosystem(AlphabeticStrings()); affineCipher
P = affineCipher.encoding("The affine cryptosystem."); P

# encrypt the plaintext using the key (3, 13)
a, b = (3, 13)
C = affineCipher.enciphering(a, b, P); C

# decrypt the ciphertext
DC = affineCipher.deciphering(a, b, C); DC

# control result
DC == P
```

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

- ▶ The **Atbash** cipher is used to obscure names of people and places in the Hebrew bible
- ▶ It consists in substituting **aleph** (the first letter) for **tav** (the last), **beth** (the second) for **shin** (one before last), and so on, reversing the alphabet.

א ב ג ד ה ז ח ט י כ ל מ נ ס ע פ צ ק ר ש ת
ת ש ר ק צ פ ע ס נ מ ל כ י ט ח ז ו ה ד נ ב א

right-to-left
left-to-right

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

- ▶ The **Atbash** cipher is used to obscure names of people and places in the **Hebrew** bible
- ▶ It consists in substituting **aleph** (the first letter) for **tav** (the last), **beth** (the second) for **shin** (one before last), and so on, reversing the alphabet.

א ב ג ד ה ז ח ט י כ ל מ נ ס ע פ צ ק ר ש ת
ת ש ר ק צ פ ע ס נ מ ל כ י ט ח ז ו ה ד נ ב א

right-to-left
left-to-right

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

- ▶ The **Atbash** cipher is used to obscure names of people and places in the **Hebrew bible**
- ▶ It consists in substituting **aleph** (the first letter) for **tav** (the last), **beth** (the second) for **shin** (one before last), and so on, reversing the alphabet.

א ב ג ד ה ז ח ט י כ ל מ נ ס ע פ צ ק ר ש ת
ת ש ר ק צ פ ע ס נ מ ל כ י ט ח ז ו ה ד נ ב א

right-to-left
left-to-right

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: EVMR ERWR ERXR

Principles

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **EVMR ERWR ERXR**

Principles

Substitution cipher

Each letter is replaced by another letter

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **EVMR ERWR ERXR**

Principles

- ▶ This is a cipher algorithm by mono-alphabetic substitution
- ▶ The Atbash cipher can be seen as a special case of the Affine cipher

→ [View slide presentation](#)

→ [View video](#)

→ [View exercise](#)

→ [View solution](#)

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **EVMR ERWR ERXR**

Principles

- ▶ This is a cipher algorithm by **mono-alphabetic substitution**
- ▶ The Atbash cipher can be seen as a special case of the **Affine cipher**
- ▶ If m is the length of the alphabet
- ▶ Ciphering: $C_c(x) = ax + b \pmod{m}$ whit $a = b = m - 1$
- ▶ Deciphering: $D_c(y) = ay - b \pmod{m}$ whith $a = b = m - 1$

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **EVMR ERWR ERXR**

Principles

- ▶ This is a cipher algorithm by **mono-alphabetic substitution**
- ▶ The Atbash cipher can be seen as a special case of the **Affine cipher**
- ▶ If m is the length of the alphabet
- ▶ Ciphering: $C_k(x) = ax + b \pmod{m}$ whit $a = b = m - 1$
- ▶ Deciphering: $D_k(y) = ay - b \pmod{m}$ whit $a = b = m - 1$

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **EVMR ERWR ERXR**

Principles

- ▶ This is a cipher algorithm by **mono-alphabetic substitution**
- ▶ The Atbash cipher can be seen as a special case of the **Affine cipher**
- ▶ If m is the length of the alphabet
- ▶ Ciphering: $C_k(x) = ax + b \pmod{m}$ whit $a = b = m - 1$
- ▶ Deciphering: $D_k(y) = ay - b \pmod{m}$ whith $a = b = m - 1$

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **EVMR ERWR ERXR**

Principles

- ▶ This is a cipher algorithm by **mono-alphabetic substitution**
- ▶ The Atbash cipher can be seen as a special case of the **Affine cipher**
- ▶ If m is the length of the alphabet
- ▶ Ciphering: $C_k(x) = ax + b \pmod{m}$ whit $a = b = m - 1$
- ▶ Deciphering: $D_k(y) = ay - b \pmod{m}$ whith $a = b = m - 1$

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **EVMR ERWR ERXR**

Principles

- ▶ This is a cipher algorithm by **mono-alphabetic substitution**
- ▶ The Atbash cipher can be seen as a special case of the **Affine cipher**
- ▶ If m is the length of the alphabet
- ▶ Ciphering: $C_k(x) = ax + b \pmod{m}$ whit $a = b = m - 1$
- ▶ Deciphering: $D_k(y) = ay - b \pmod{m}$ whith $a = b = m - 1$

Craftsmanship age: from the origins to WW I

The **Atbash** cipher

The Atbash cipher

The first letter of the alphabet is substituted with the last letter, the second letter for the second last and so on.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **EVMR ERWR ERXR**

Principles

- ▶ This is a cipher algorithm by **mono-alphabetic substitution**
- ▶ The Atbash cipher can be seen as a special case of the **Affine cipher**
- ▶ If m is the length of the alphabet
- ▶ Ciphering: $C_k(x) = ax + b \pmod{m}$ whit $a = b = m - 1$
- ▶ Deciphering: $D_k(y) = ay - b \pmod{m}$ whith $a = b = m - 1$

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

The Lacedaemonian Scytale

After having wound the belt on the scytale, the message was written by placing a letter on each circumvolution. The fixed diameter of the stick is the key.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: VNEIV IID IV Key: [0, 2, 1]
- ▶ It's an anagram of the original message

Principles

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

The Lacedaemonian Scytale

After having wound the belt on the scytale, the message was written by placing a letter on each circumvolution. The fixed diameter of the stick is the key.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **VNEIV IID IV** Key: [0, 2, 1]
- ▶ It's an **anagram** of the original message

Principles

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

The Lacedaemonian Scytale

After having wound the belt on the scytale, the message was written by placing a letter on each circumvolution. The fixed diameter of the stick is the key.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **VNEIV IID IV** Key: **[0, 2, 1]**
- ▶ It's an **anagram** of the original message

Principles

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

The Lacedaemonian Scytale

After having wound the belt on the scytale, the message was written by placing a letter on each circumvolution. The fixed diameter of the stick is the key.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **VNEIV IID IV** Key: **[0, 2, 1]**
- ▶ It's an **anagram** of the original message

Principles

- ▶ This is a cipher algorithm by permutation
- ▶ Mapping: $(0, 1, 2, \dots, n-1) \rightarrow (0, 1, 2, \dots, n-1)$
- ▶ $n > m$

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

The Lacedaemonian Scytale

After having wound the belt on the scytale, the message was written by placing a letter on each circumvolution. The fixed diameter of the stick is the key.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: VNEIV IID IV Key: [0, 2, 1]
- ▶ It's an anagram of the original message

Principles

- ▶ This is a cipher algorithm by permutation
- ▶ Ciphering: $C_k(x_1, \dots, x_m) = (x_{k(1)}, x_{k(2)}, \dots, x_{k(m)})$
- ▶ Deciphering: $D_k(y_1, \dots, y_m) = (y_{k(1)}^{-1}, y_{k(2)}^{-1}, \dots, y_{k(m)}^{-1})$

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

The Lacedaemonian Scytale

After having wound the belt on the scytale, the message was written by placing a letter on each circumvolution. The fixed diameter of the stick is the key.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: VNEIV IID IV Key: [0, 2, 1]
- ▶ It's an anagram of the original message

Principles

- ▶ This is a cipher algorithm by permutation
- ▶ Ciphering: $C_k(x_1, \dots, x_m) = (x_{k(1)}, x_{k(2)}, \dots, x_{k(m)})$
- ▶ Deciphering: $D_k(y_1, \dots, y_m) = (y_{k(1)}^{-1}, y_{k(2)}^{-1}, \dots, y_{k(m)}^{-1})$

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

The Lacedaemonian Scytale

After having wound the belt on the scytale, the message was written by placing a letter on each circumvolution. The fixed diameter of the stick is the key.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: VNEIV IID IV Key: [0, 2, 1]
- ▶ It's an anagram of the original message

Principles

- ▶ This is a cipher algorithm by permutation
- ▶ Ciphering: $C_k(x_1, \dots, x_m) = (x_{k(1)}, x_{k(2)}, \dots, x_{k(m)})$
- ▶ Deciphering: $D_k(y_1, \dots, y_m) = (y_{k(1)}^{-1}, y_{k(2)}^{-1}, \dots, y_{k(m)}^{-1})$

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

The Lacedaemonian Scytale

After having wound the belt on the scytale, the message was written by placing a letter on each circumvolution. The fixed diameter of the stick is the key.

- ▶ Message: VENI VIDI VICI
- ▶ Cryptogram: VNEIV IID IV Key: [0, 2, 1]
- ▶ It's an anagram of the original message

Principles

- ▶ This is a cipher algorithm by permutation
- ▶ Ciphering: $C_k(x_1, \dots, x_m) = (x_{k(1)}, x_{k(2)}, \dots, x_{k(m)})$
- ▶ Deciphering: $D_k(y_1, \dots, y_m) = (y_{k(1)}^{-1}, y_{k(2)}^{-1}, \dots, y_{k(m)}^{-1})$

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

Block cipher

This method of encryption introduces the concept of **blocks**: the message is divided into blocks of the length of the permutation key. Each block is encrypted independently.



Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

Encryption by permutation: visualization



Key length = 4
Key length = 10

Key length = 40
Key length = 100

Key length = 400
Key length = 4000

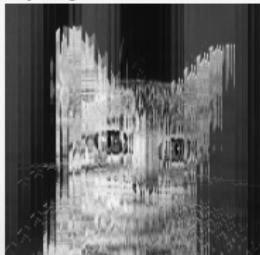
Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

Encryption by permutation: visualization



Key length = 4



Key length = 10

Key length = 40
Key length = 100

Key length = 400
Key length = 4000

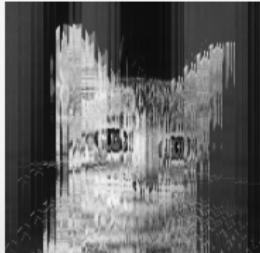
Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

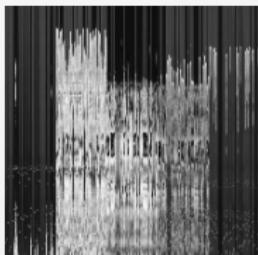
Encryption by permutation: visualization



Key length = 4



Key length = 10



Key length = 40
Key length = 100

Key length = 400
Key length = 4000

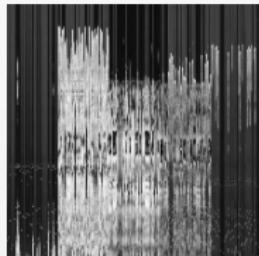
Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

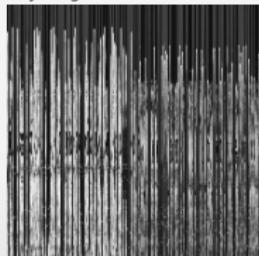
Encryption by permutation: visualization



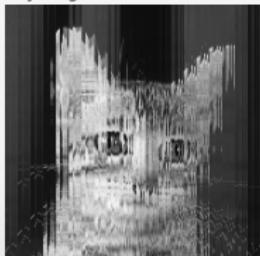
Key length = 4



Key length = 40



Key length = 100



Key length = 10

Key length = 400
Key length = 4000

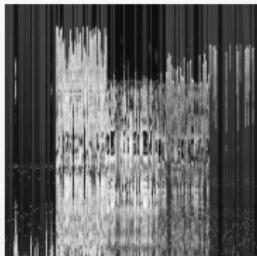
Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

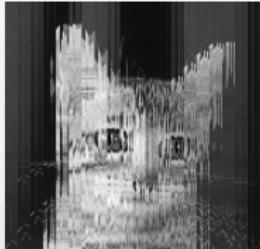
Encryption by permutation: visualization



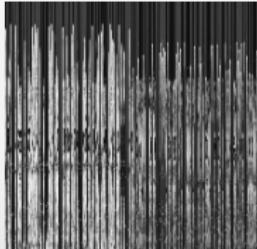
Key length = 4



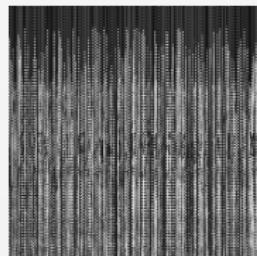
Key length = 40



Key length = 10



Key length = 100



Key length = 400
Key length = 4000

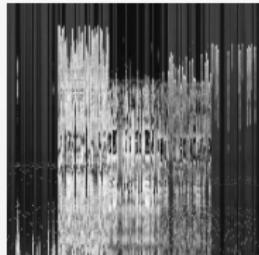
Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

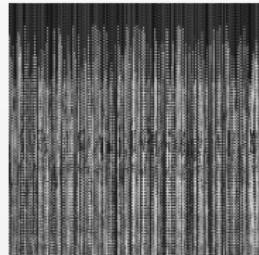
Encryption by permutation: visualization



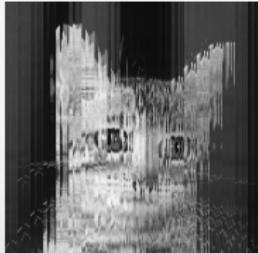
Key length = 4



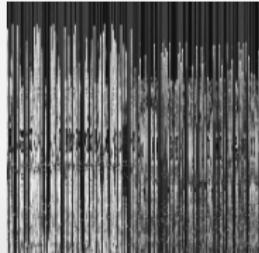
Key length = 40



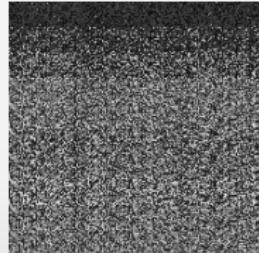
Key length = 400



Key length = 10



Key length = 100



Key length = 4000

Craftsmanship age: from the origins to WW I

The Lacedaemonian Scytale

Practice with Sage: **Permutation**

```
# transposition cipher using a block length of 16
A = AlphabeticStrings()
T = TranspositionCryptosystem(A, 16); T
P = "The sky is blue it will be fine tomorrow"; P
K = T.random_key(); K

# encode plaintext
msg = T.encoding(P)

# ciphering
C = T.enciphering(K, msg); C

# deciphering
DC = T.deciphering(K, C); DC

# control result
DC == msg
```

Craftsmanship age: from the origins to WW I

The Vigenère cipher

The Vigenère cipher

In the sixteenth century, **Blaise de Vigenère**, French diplomat, imagined an extension of the Caesar cipher by adding the concept of key. The key defined a variable offset for each letter of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **QMTMMZZHDHBMPM** Key: *vigenere*

Principles

Craftsmanship age: from the origins to WW I

The Vigenère cipher

The Vigenère cipher

In the sixteenth century, **Blaise de Vigenère**, French diplomat, imagined an extension of the Caesar cipher by adding the concept of key. The key defined a variable offset for each letter of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **QMTMMZZHDHBMPM** Key: *vigenere*

Principles

Craftsmanship age: from the origins to WW I

The Vigenère cipher

The Vigenère cipher

In the sixteenth century, **Blaise de Vigenère**, French diplomat, imagined an extension of the Caesar cipher by adding the concept of key. The key defined a variable offset for each letter of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **QMTMMZZHDHBMPM** Key: *vigenere*

Principles

- ▶ This is a cipher algorithm by poly-alphabetic substitution
- ▶ Cryptogram: QMTMMZZHDHBMPM
- ▶ Key: vigenere

Craftsmanship age: from the origins to WW I

The Vigenère cipher

The Vigenère cipher

In the sixteenth century, **Blaise de Vigenère**, French diplomat, imagined an extension of the Caesar cipher by adding the concept of key. The key defined a variable offset for each letter of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **QMTMMZZHDHBMPM** Key: *vigenere*

Principles

- ▶ This is a cipher algorithm by **poly-alphabetic substitution**
- ▶ Ciphering: $C_k(x_1, \dots, x_n) = (x_1 + k_1 \text{ mod } 26), \dots, (x_n + k_n \text{ mod } 26)$
- ▶ Deciphering: $D_k(y_1, \dots, y_n) = (y_1 - k_1 \text{ mod } 26), \dots, (y_n - k_n \text{ mod } 26)$

Craftsmanship age: from the origins to WW I

The Vigenère cipher

The Vigenère cipher

In the sixteenth century, **Blaise de Vigenère**, French diplomat, imagined an extension of the Caesar cipher by adding the concept of key. The key defined a variable offset for each letter of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **QMTMMZZHDHBMPM** Key: *vigenere*

Principles

- ▶ This is a cipher algorithm by **poly-alphabetic substitution**
- ▶ Ciphering: $C_k(x_1, \dots, x_n) = (x_1 + k_1 \text{ mod } 26), \dots, (x_n + k_n \text{ mod } 26)$
- ▶ Deciphering: $D_k(y_1, \dots, y_n) = (y_1 - k_1 \text{ mod } 26), \dots, (y_n - k_n \text{ mod } 26)$

Craftsmanship age: from the origins to WW I

The Vigenère cipher

The Vigenère cipher

In the sixteenth century, **Blaise de Vigenère**, French diplomat, imagined an extension of the Caesar cipher by adding the concept of key. The key defined a variable offset for each letter of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **QMTMMZZHDHBMPM** Key: *vigenere*

Principles

- ▶ This is a cipher algorithm by **poly-alphabetic substitution**
- ▶ Ciphering: $C_k(x_1, \dots, x_n) = (x_1 + k_1 \text{ mod } 26), \dots, (x_n + k_n \text{ mod } 26)$
- ▶ Deciphering: $D_k(y_1, \dots, y_n) = (y_1 - k_1 \text{ mod } 26), \dots, (y_n - k_n \text{ mod } 26)$

Craftsmanship age: from the origins to WW I

The Vigenère cipher

The Vigenère cipher

In the sixteenth century, **Blaise de Vigenère**, French diplomat, imagined an extension of the Caesar cipher by adding the concept of key. The key defined a variable offset for each letter of the message.

- ▶ Message: **VENI VIDI VICI**
- ▶ Cryptogram: **QMTMMZZHDHBMPM** Key: *vigenere*

Principles

- ▶ This is a cipher algorithm by **poly-alphabetic substitution**
- ▶ Ciphering: $C_k(x_1, \dots, x_n) = (x_1 + k_1 \text{ mod } 26), \dots, (x_n + k_n \text{ mod } 26)$
- ▶ Deciphering: $D_k(y_1, \dots, y_n) = (y_1 - k_1 \text{ mod } 26), \dots, (y_n - k_n \text{ mod } 26)$

Craftsmanship age: from the origins to WW I

The Vigenère cipher

The Vigenère square

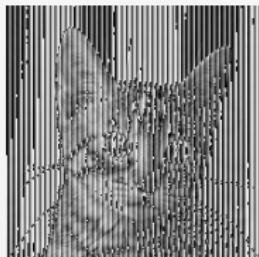
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y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



Craftsmanship age: from the origins to WW I

The Vigenère cipher

Encryption by poly-alphabetic substitution: visualization



Key length = 4
Key length = 10

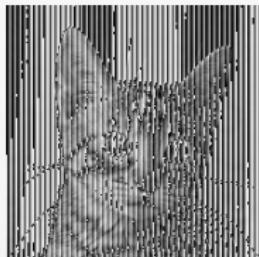
Key length = 40
Key length = 100

Key length = 400
Key length = 4000

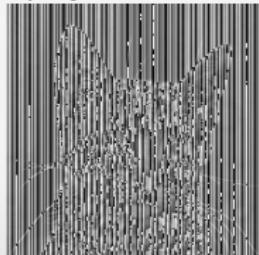
Craftsmanship age: from the origins to WW I

The Vigenère cipher

Encryption by poly-alphabetic substitution: visualization



Key length = 4



Key length = 10

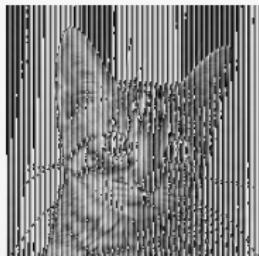
Key length = 40
Key length = 100

Key length = 400
Key length = 4000

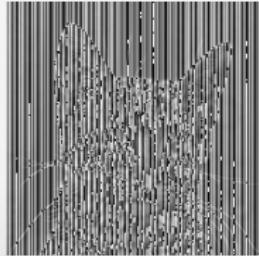
Craftsmanship age: from the origins to WW I

The Vigenère cipher

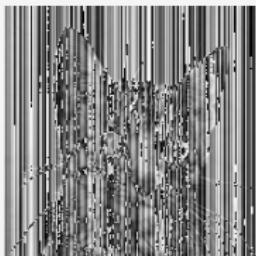
Encryption by poly-alphabetic substitution: visualization



Key length = 4



Key length = 10



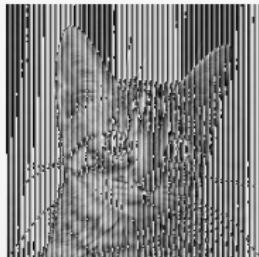
Key length = 40
Key length = 100

Key length = 400
Key length = 4000

Craftsmanship age: from the origins to WW I

The Vigenère cipher

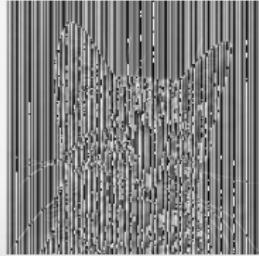
Encryption by poly-alphabetic substitution: visualization



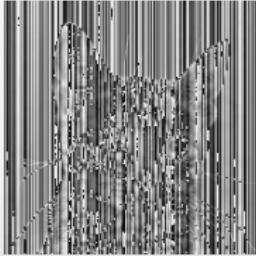
Key length = 4



Key length = 40



Key length = 10



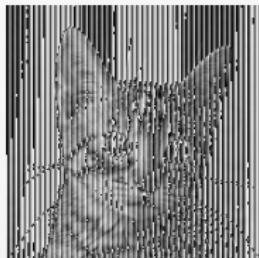
Key length = 100

Key length = 400
Key length = 4000

Craftsmanship age: from the origins to WW I

The Vigenère cipher

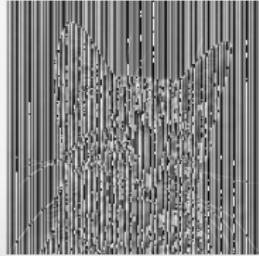
Encryption by poly-alphabetic substitution: visualization



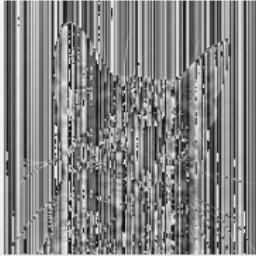
Key length = 4



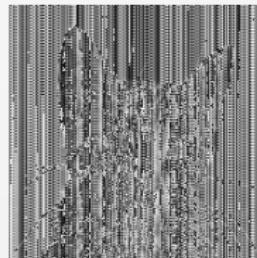
Key length = 40



Key length = 10



Key length = 100

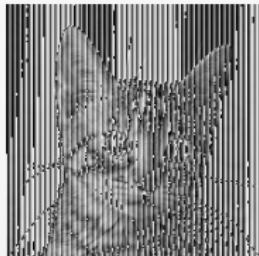


Key length = 400
Key length = 4000

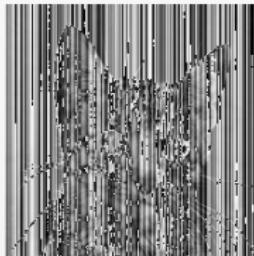
Craftsmanship age: from the origins to WW I

The Vigenère cipher

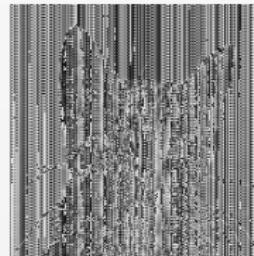
Encryption by poly-alphabetic substitution: visualization



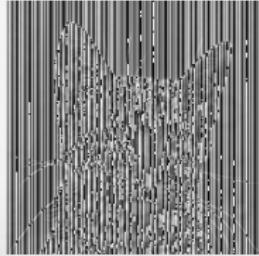
Key length = 4



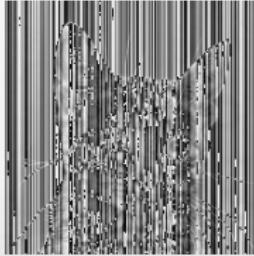
Key length = 40



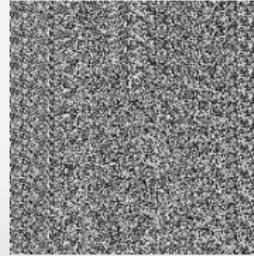
Key length = 400



Key length = 10



Key length = 100



Key length = 4000

Craftsmanship age: from the origins to WW I

The Vigenère cipher

Practice with Sage: **Vigenère cipher**

```
# construct Vigenere cipher
keylen = 14
A = AlphabeticStrings()
V = VigenereCryptosystem(A, keylen); V
key = V.random_key(); key # alternative key = A('ABCDEFGHIJKLMN')
len(key)

# encoding
P = "The Vigenere cipher is polyalphabetic."
msg = V.encoding(P); msg # alternative: msg = A.encoding(P); msg

# encryption
C = V.enciphering(key, msg); C

# decryption
DC = V.deciphering(key, C); DC

# control result
msg == DC
```

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle



Auguste Kerckhoffs

- ▶ born January 19, 1835
- ▶ died August 9, 1903
- ▶ Dutch
- ▶ Doctor of Letters
- ▶ German teacher at *École des Hautes Études Commerciales*
- ▶ linguist and cryptographer
- ▶ defender of the language *volapük*

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle



Auguste Kerckhoffs

- ▶ born January 19, 1835
- ▶ died August 9, 1903
- ▶ Dutch
- ▶ Doctor of Letters
- ▶ German teacher at *École des Hautes Études Commerciales*
- ▶ linguist and cryptographer
- ▶ defender of the language *volapük*

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle



Auguste Kerckhoffs

- ▶ born January 19, 1835
- ▶ died August 9, 1903
- ▶ Dutch
- ▶ Doctor of Letters
- ▶ German teacher at *École des Hautes Études Commerciales*
- ▶ linguist and cryptographer
- ▶ defender of the language *volapük*

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle



Auguste Kerckhoffs

- ▶ born January 19, 1835
- ▶ died August 9, 1903
- ▶ Dutch
- ▶ Doctor of Letters
- ▶ German teacher at *École des Hautes Études Commerciales*
- ▶ linguist and cryptographer
- ▶ defender of the language *volapük*

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle



Auguste Kerckhoffs

- ▶ born January 19, 1835
- ▶ died August 9, 1903
- ▶ Dutch
- ▶ Doctor of Letters
- ▶ German teacher at *École des Hautes Études Commerciales*
- ▶ linguist and cryptographer
- ▶ defender of the language *volapük*

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle



Auguste Kerckhoffs

- ▶ born January 19, 1835
- ▶ died August 9, 1903
- ▶ Dutch
- ▶ Doctor of Letters
- ▶ German teacher at *École des Hautes Études Commerciales*
- ▶ linguist and cryptographer
- ▶ defender of the language *volapük*

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle



Auguste Kerckhoffs

- ▶ born January 19, 1835
- ▶ died August 9, 1903
- ▶ Dutch
- ▶ Doctor of Letters
- ▶ German teacher at *École des Hautes Études Commerciales*
- ▶ linguist and cryptographer
- ▶ defender of the language *volapük*

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Auguste Kerckhoffs describes the principles of modern cryptography in his essay "**The Military Cryptography**" published in 1883 in the "Journal of Military Science"

- ➊ The system must be practically, if not mathematically, indecipherable
- ➋ It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ➌ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents
- ➍ It must be applicable to telegraphic correspondence
- ➎ It must be portable, and its usage and function must not require the concourse of several people
- ➏ Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Auguste Kerckhoffs describes the principles of modern cryptography in his essay "**The Military Cryptography**" published in 1883 in the "Journal of Military Science"

- ➊ The system must be practically, if not mathematically, indecipherable
- ➋ It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ➌ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents
- ➍ It must be applicable to telegraphic correspondence
- ➎ It must be portable, and its usage and function must not require the concourse of several people
- ➏ Finally, it is necessary, given the circumstances that command its application, that the system be easy to learn, requiring neither mental strain nor the knowledge of a long series of

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Auguste Kerckhoffs describes the principles of modern cryptography in his essay "**The Military Cryptography**" published in 1883 in the "Journal of Military Science"

- ➊ The system must be practically, if not mathematically, indecipherable
- ➋ It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ➌ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents
- ➍ It must be applicable to telegraphic correspondence
- ➎ It must be portable, and its usage and function must not require the concourse of several people
- ➏ Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Auguste Kerckhoffs describes the principles of modern cryptography in his essay "**The Military Cryptography**" published in 1883 in the "Journal of Military Science"

- ➊ The system must be practically, if not mathematically, indecipherable
- ➋ It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ➌ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents
- ➍ It must be applicable to telegraphic correspondence
- ➎ It must be portable, and its usage and function must not require the concourse of several people
- ➏ Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Auguste Kerckhoffs describes the principles of modern cryptography in his essay "**The Military Cryptography**" published in 1883 in the "Journal of Military Science"

- ① The system must be practically, if not mathematically, indecipherable
- ② It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ③ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents
- ④ It must be applicable to telegraphic correspondence
- ⑤ It must be portable, and its usage and function must not require the concourse of several people
- ⑥ Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Auguste Kerckhoffs describes the principles of modern cryptography in his essay "**The Military Cryptography**" published in 1883 in the "Journal of Military Science"

- ① The system must be practically, if not mathematically, indecipherable
- ② It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ③ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents
- ④ It must be applicable to telegraphic correspondence
- ⑤ It must be portable, and its usage and function must not require the concourse of several people
- ⑥ Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Comments on the principles of Kerckhoffs

- ① The system must be practically, if not mathematically, indecipherable
- ② It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ③ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents

Points 2 and 3 are the new fundamental axioms of cryptography

- ▶ The cipher algorithm is **public**
- ▶ the secret lies in the encryption key
- ▶ a cryptographic algorithm based solely on the secrecy of the algorithm is dangerous because the day where it is discovered his security model collapses

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Comments on the principles of Kerckhoffs

- ① The system must be practically, if not mathematically, indecipherable
- ② It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ③ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents

Points 2 and 3 are the new fundamental axioms of cryptography

- ▶ The cipher algorithm is **public**
- ▶ the secret lies in the encryption key
- ▶ a cryptographic algorithm based solely on the secrecy of the algorithm is dangerous because the day where it is discovered his security model collapses

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Comments on the principles of Kerckhoffs

- ① The system must be practically, if not mathematically, indecipherable
- ② It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ③ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents

Points 2 and 3 are the new fundamental axioms of cryptography

- ▶ The cipher algorithm is **public**
- ▶ the secret lies in the encryption key
- ▶ a cryptographic algorithm based solely on the secrecy of the algorithm is dangerous because the day where it is discovered his security model collapses

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Comments on the principles of Kerckhoffs

- ① The system must be practically, if not mathematically, indecipherable
- ② It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience
- ③ Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents

Points 2 and 3 are the new fundamental axioms of cryptography

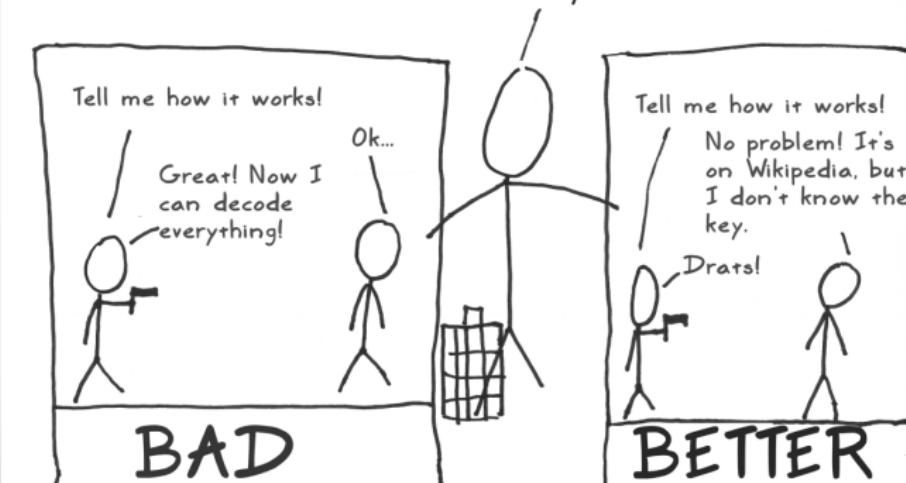
- ▶ The cipher algorithm is **public**
- ▶ the secret lies in the encryption key
- ▶ a cryptographic algorithm based solely on the secrecy of the algorithm is dangerous because the day where it is discovered his security model collapses

Craftsmanship age: from the origins to WW I

Kerckhoffs's principle

Big Idea #3: Secrecy Only in the Key

After thousands of years, we learned
that it's a bad idea to assume that no
one knows how your method works.
Someone will eventually find that out.



Craftsmanship age: from the origins to WW I

Conclusion: craftsmanship age

"There is no cipher unbreakable with forty page of an encrypted telegram."

Napoléon



- ➊ The cryptanalysts have the advantage
- ➋ Means of encryption limited
- ➌ Emergence of the concept of key

Fundamental discovery

The secret do not lies in the cipher algorithm but in the settings of this algorithm: *the cipher key*.

Craftsmanship age: from the origins to WW I

Conclusion: craftsmanship age

"There is no cipher unbreakable with forty page of an encrypted telegram."

Napoléon



- ① The cryptanalysts have the advantage
- ② Means of encryption limited
- ③ Emergence of the concept of key

Fundamental discovery

The secret do not lies in the cipher algorithm but in the settings of this algorithm: *the cipher key*.

Craftsmanship age: from the origins to WW I

Conclusion: craftsmanship age

"There is no cipher unbreakable with forty page of an encrypted telegram."

Napoléon



- ① The cryptanalysts have the advantage
- ② Means of encryption limited
- ③ Emergence of the concept of key

Fundamental discovery

The **secret** do not lies in the cipher algorithm but in the settings of this algorithm: *the cipher key*.

Craftsmanship age: from the origins to WW I

Conclusion: craftsmanship age

"There is no cipher unbreakable with forty page of an encrypted telegram."

Napoléon



- ① The cryptanalysts have the advantage
- ② Means of encryption limited
- ③ Emergence of the concept of key

Fundamental discovery

The **secret** do not lies in the cipher algorithm but in the settings of this algorithm: *the cipher key*.

Craftsmanship age: from the origins to WW I

Conclusion: craftsmanship age

"There is no cipher unbreakable with forty page of an encrypted telegram."

Napoléon



- ① The cryptanalysts have the advantage
- ② Means of encryption limited
- ③ Emergence of the concept of key

Fundamental discovery

The **secret** do not lies in the cipher algorithm but in the settings of this algorithm: *the cipher key*.

Second period – technical age: from WWI to Shannon's advent

3 History

- First period – craftsmanship age: from the origins to WW I
- Second period – technical age: from WWI to Shannon's advent
- Third period – modern age: from the end of WW II to our days



Technical age: from WWI to Shannon's advent

ADFGVX cipher

The **ADFGVX** cipher was used by the German Army during World War I

- ▶ **ADFGVX** is an extension of an earlier cipher called **ADFGX**
- ▶ **ADFGX** encrypt 25 characters (*i* and *j* are combined) and **ADFGVX** encrypt 36 characters (*letters plus numbers*)
- ▶ This cipher was invented by **Colonel Fritz Nebel**
- ▶ This cipher was restricted to German High Command communications between and among the headquarters of divisions and army corps
- ▶ The cipher is named after the six possible letters used in the ciphertext: A, D, F, G, V and X
- ▶ These particular letters were chosen for their distinctive Morse encoding to prevent errors in transmission

Technical age: from WWI to Shannon's advent

ADFGVX cipher

The **ADFGVX** cipher was used by the German Army during World War I

- ▶ **ADFGVX** is an extension of an earlier cipher called **ADFGX**
- ▶ **ADFGX** encrypt 25 characters (*i and j are combined*) and **ADFGVX** encrypt 36 characters (*letters plus numbers*)
- ▶ This cipher was invented by **Colonel Fritz Nebel**
- ▶ This cipher was restricted to German High Command communications between and among the headquarters of divisions and army corps
- ▶ The cipher is named after the six possible letters used in the ciphertext: **A, D, F, G, V and X**
- ▶ These particular letters were chosen for their distinctive Morse encoding to prevent errors in transmission

Technical age: from WWI to Shannon's advent

ADFGVX cipher

The ADFGVX cipher was used by the German Army during World War I

- ▶ ADFGVX is an extension of an earlier cipher called ADFGX
- ▶ ADFGX encrypt 25 characters (*i and j are combined*) and ADFGVX encrypt 36 characters (*letters plus numbers*)
- ▶ This cipher was invented by Colonel Fritz Nebel
- ▶ This cipher was restricted to German High Command communications between and among the headquarters of divisions and army corps
- ▶ The cipher is named after the six possible letters used in the ciphertext: A, D, F, G, V and X
- ▶ These particular letters were chosen for their distinctive Morse encoding to prevent errors in transmission

Technical age: from WWI to Shannon's advent

ADFGVX cipher

The ADFGVX cipher was used by the German Army during World War I

- ▶ ADFGVX is an extension of an earlier cipher called ADFGX
- ▶ ADFGX encrypt 25 characters (*i and j are combined*) and ADFGVX encrypt 36 characters (*letters plus numbers*)
- ▶ This cipher was invented by Colonel Fritz Nebel
- ▶ This cipher was restricted to German High Command communications between and among the headquarters of divisions and army corps
- ▶ The cipher is named after the six possible letters used in the ciphertext: A, D, F, G, V and X
- ▶ These particular letters were chosen for their distinctive Morse encoding to prevent errors in transmission

Technical age: from WWI to Shannon's advent

ADFGVX cipher

The ADFGVX cipher was used by the German Army during World War I

- ▶ ADFGVX is an extension of an earlier cipher called ADFGX
- ▶ ADFGX encrypt 25 characters (*i and j are combined*) and ADFGVX encrypt 36 characters (*letters plus numbers*)
- ▶ This cipher was invented by Colonel Fritz Nebel
- ▶ This cipher was restricted to German High Command communications between and among the headquarters of divisions and army corps
- ▶ The cipher is named after the six possible letters used in the ciphertext: A, D, F, G, V and X
- ▶ These particular letters were chosen for their distinctive Morse encoding to prevent errors in transmission

Technical age: from WWI to Shannon's advent

ADFGVX cipher

The ADFGVX cipher was used by the German Army during World War I

- ▶ ADFGVX is an extension of an earlier cipher called ADFGX
- ▶ ADFGX encrypt 25 characters (*i and j are combined*) and ADFGVX encrypt 36 characters (*letters plus numbers*)
- ▶ This cipher was invented by Colonel Fritz Nebel
- ▶ This cipher was restricted to German High Command communications between and among the headquarters of divisions and army corps
- ▶ The cipher is named after the six possible letters used in the ciphertext: A, D, F, G, V and X
- ▶ These particular letters were chosen for their distinctive Morse encoding to prevent errors in transmission

Technical age: from WWI to Shannon's advent

ADFGVX cipher

- ▶ a decryption feat was achieved by French **Captain Georges Painvin**
- ▶ He broke the German system ADFGVX in April 1918
- ▶ *Before they launched their last offensive, the Germans modified the system, and in a few days Painvin broke it once more. The French then discovered where the Germans wanted to attack, and could stop the offensive.*



Technical age: from WWI to Shannon's advent

ADFGVX cipher

- ▶ a decryption feat was achieved by French **Captain Georges Painvin**
- ▶ He broke the German system ADFGVX in April 1918
- ▶ *Before they launched their last offensive, the Germans modified the system, and in a few days Painvin broke it once more. The French then discovered where the Germans wanted to attack, and could stop the offensive.*



Technical age: from WWI to Shannon's advent

ADFGVX cipher

- ▶ a decryption feat was achieved by French **Captain Georges Painvin**
- ▶ He broke the German system ADFGVX in April 1918
- ▶ *Before they launched their last offensive, the Germans modified the system, and in a few days Painvin broke it once more. The French then discovered where the Germans wanted to attack, and could stop the offensive.*



Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in the grid
 - ▶ Read the final encoding from the table in a column-major order

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

- ▶ Read the final encoding from the table in a column-major order
- ▶ Finally add spaces to aid readability

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Read the final encoding from the table in a column-major order
 - ▶ Finally add spaces to aid readability

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Choose a keyword
 - ▶ Place the encoded words in a table with as many columns as the key word using row-major ordering
 - ▶ Rearrange columns by the alphabetical order of the key word's letters
 - ▶ Column-major order
 - ▶ Finally add spaces to aid readability

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Choose a keyword
 - ▶ Place the encoded words in a table with as many columns as the key word using row-major ordering
 - ▶ Rearrange columns by the alphabetical order of the key word's letters
 - ▶ Column-major order
 - ▶ Finally add spaces to aid readability

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Choose a keyword
 - ▶ Place the encoded words in a table with as many columns as the key word using row-major ordering
 - ▶ Rearrange columns by the alphabetical order of the key word's letters
 - ▶ column-major order
 - ▶ Finally add spaces to aid readability

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Choose a keyword
 - ▶ Place the encoded words in a table with as many columns as the key word using row-major ordering
 - ▶ Rearrange columns by the alphabetical order of the key word's letters
 - ▶ Read the final encoding from the table in a column-major order
 - ▶ Finally add spaces to aid readability

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Choose a keyword
 - ▶ Place the encoded words in a table with as many columns as the key word using row-major ordering
 - ▶ Rearrange columns by the alphabetical order of the key word's letters
 - ▶ Read the final encoding from the table in a column-major order

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Choose a keyword
 - ▶ Place the encoded words in a table with as many columns as the key word using row-major ordering
 - ▶ Rearrange columns by the alphabetical order of the key word's letters
 - ▶ Read the final encoding from the table in a column-major order
 - ▶ Finally add spaces to aid readability

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Choose a keyword
 - ▶ Place the encoded words in a table with as many columns as the key word using row-major ordering
 - ▶ Rearrange columns by the alphabetical order of the key word's letters
 - ▶ Read the final encoding from the table in a column-major order
 - ▶ Finally add spaces to aid readability

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Principles of the ADFGVX cipher

- ▶ ADFGVX is a **fractionating transposition cipher**
- ▶ It combines a modified Polybius square with a single columnar transposition
- ▶ Mechanism of encryption
 - ▶ Remove spaces of the plain text
 - ▶ Use the ADFGVX grid to replace letters of plain text with a pair of letters that correspond to the row and column of the letter in a grid
 - ▶ Choose a keyword
 - ▶ Place the encoded words in a table with as many columns as the key word using row-major ordering
 - ▶ Rearrange columns by the alphabetical order of the key word's letters
 - ▶ Read the final encoding from the table in a column-major order
 - ▶ Finally add spaces to aid readability

A	D	F	G	V	X
A	8	P	3	D	1
D	L	T	4	O	A
F	7	K	B	C	5
G	J	U	6	W	G
V	X	S	V	I	R
X	9	E	Y	0	F
					Q

ADFGVX Encoding Grid

Fractionation: Generic term for all kinds of methods that encrypt one plain text character by several cipher text characters and then apply a transposition cipher to this cipher text so that cipher text characters originally belonging to each other are separated.

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Operation of ADFGVX

- ▶ Plain text: **Attack tomorrow**
- ▶ Removing spaces: **Attacktomorrow**
- ▶ Using the square for the first encryption step:

A	T	T	A	C	K	T	O	M	O	R	R	O	W
DV	DD	DD	DV	FG	FD	DD	DG	GX	DG	VV	VV	DG	GG

- ▶ Writing the message in rows under the transposition key, **ENCRYPT**:

E	N	C	R	Y	P	T
D	V	D	D	D	D	D
V	F	G	F	D	D	D
D	G	G	X	D	G	Y
V	V	V	D	G	G	G

so ENCRYPT becomes CENPRTY

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Operation of ADFGVX

- ▶ Plain text: **Attack tomorrow**
- ▶ Removing spaces: **Attacktomorrow**
- ▶ Using the square for the first encryption step:

A	T	T	A	C	K	T	O	M	O	R	R	O	W
DV	DD	DD	DV	FG	FD	DD	DG	GX	DG	VV	VV	DG	GG

- ▶ Writing the message in rows under the transposition key, **ENCRYPT**:

E	N	C	R	Y	P	T
D	V	D	D	D	D	D
V	F	G	F	D	D	D
D	G	G	X	D	G	V
V	V	V	D	G	G	G

- ▶ Sorting the letters alphabetically in the transposition key,
ENCRYPT → **CENPRTY**

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Operation of ADFGVX

- ▶ Plain text: **Attack tomorrow**
- ▶ Removing spaces: **Attacktomorrow**
- ▶ Using the square for the first encryption step:

A	T	T	A	C	K	T	O	M	O	R	R	O	W
DV	DD	DD	DV	FG	FD	DD	DG	GX	DG	VV	VV	DG	GG

- ▶ Writing the message in rows under the transposition key, **ENCRYPT**:

E	N	C	R	Y	P	T
D	V	D	D	D	D	D
V	F	G	F	D	D	D
D	G	G	X	D	G	V
V	V	V	D	G	G	G

- ▶ Sorting the letters alphabetically in the transposition key,
so **ENCRYPT** becomes **CENPRTY**

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Operation of ADFGVX

- ▶ Plain text: **Attack tomorrow**
- ▶ Removing spaces: **Attacktomorrow**
- ▶ Using the square for the first encryption step:

A	T	T	A	C	K	T	O	M	O	R	R	O	W
DV	DD	DD	DV	FG	FD	DD	DG	GX	DG	VV	VV	DG	GG

- ▶ Writing the message in rows under the transposition key, **ENCRYPT**:

E	N	C	R	Y	P	T
D	V	D	D	D	D	D
V	F	G	F	D	D	D
D	G	G	X	D	G	V
V	V	V	D	G	G	G

- ▶ Sorting the letters alphabetically in the transposition key,
so **ENCRYPT** becomes **CENPRTY**

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Operation of ADFGVX

- ▶ Plain text: **Attack tomorrow**
- ▶ Removing spaces: **Attacktomorrow**
- ▶ Using the square for the first encryption step:

A	T	T	A	C	K	T	O	M	O	R	R	O	W
DV	DD	DD	DV	FG	FD	DD	DG	GX	DG	VV	VV	DG	GG

- ▶ Writing the message in rows under the transposition key, **ENCRYPT**:

E	N	C	R	Y	P	T
D	V	D	D	D	D	D
V	F	G	F	D	D	D
D	G	G	X	D	G	V
V	V	V	D	G	G	G

- ▶ Sorting the letters alphabetically in the transposition key,
so **ENCRYPT** becomes **CENPRTY**

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Operation of ADFGVX

- ▶ Rearranging the columns beneath the letters along with the letters themselves:

C	E	N	P	R	T	Y
D	D	V	D	D	D	D
G	V	F	D	F	D	D
G	D	G	G	X	V	D
V	V	V	G	D	G	G

- ▶ Reading off the text in columns and adding spaces to aid readability

DGGVD VDVVF GVDDG GDFXD DDVGD DDG

Technical age: from WWI to Shannon's advent

ADFGVX cipher

Operation of ADFGVX

- ▶ Rearranging the columns beneath the letters along with the letters themselves:

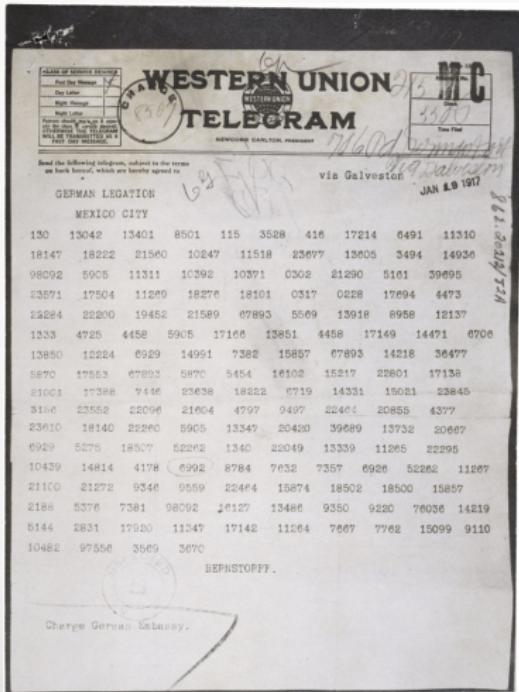
C	E	N	P	R	T	Y
D	D	V	D	D	D	D
G	V	F	D	F	D	D
G	D	G	G	X	V	D
V	V	V	G	D	G	G

- ▶ Reading off the text in columns and adding spaces to aid readability

DGGVD VDVVF GVDDG GDFXD DDVGD DDG

Technical age: from WWI to Shannon's advent

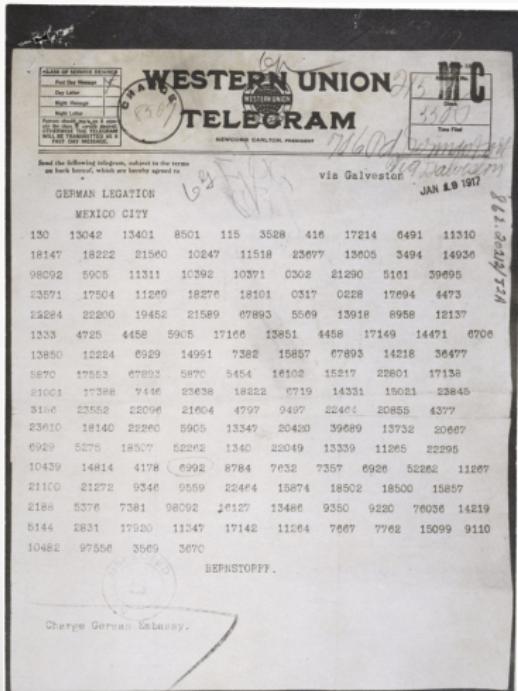
The **Zimmermann** telegram



- ▶ It was a 1917 diplomatic proposal from the **German Empire to Mexico** to make war against the **United States**
- ▶ The proposal was **caught** by the British before it could get to Mexico
- ▶ It was intercepted and decoded by the **British cryptographers of Room 40**
- ▶ The revelation angered the Americans and led in part to a U.S. declaration of war in April

Technical age: from WWI to Shannon's advent

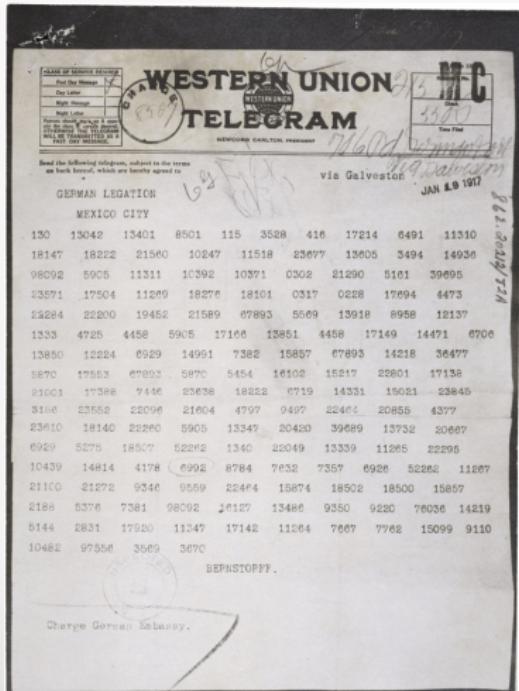
The **Zimmermann** telegram



- ▶ It was a 1917 diplomatic proposal from the **German Empire** to **Mexico** to make war against the **United States**
- ▶ The proposal was **caught** by the **British** before it could get to Mexico
- ▶ It was intercepted and decoded by the **British** cryptographers of **Room 40**
- ▶ The revelation angered the Americans and led in part to a U.S. declaration of war in April

Technical age: from WWI to Shannon's advent

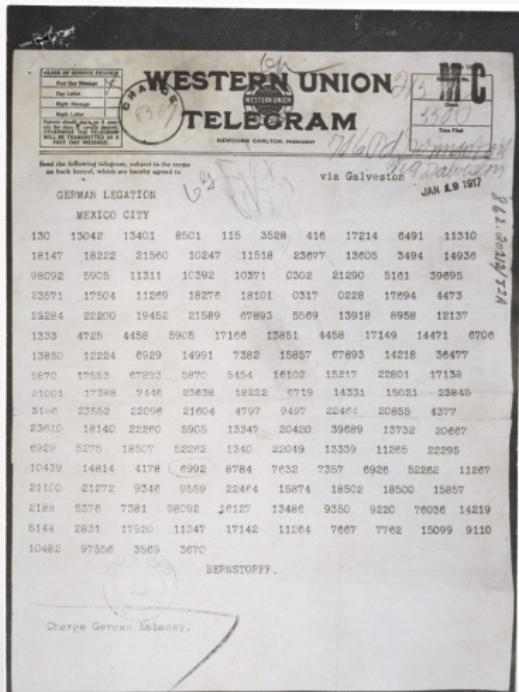
The **Zimmermann** telegram



- ▶ It was a 1917 diplomatic proposal from the **German Empire** to **Mexico** to make war against the **United States**
- ▶ The proposal was **caught** by the **British** before it could get to Mexico
- ▶ It was intercepted and decoded by the **British** cryptographers of **Room 40**
- ▶ The revelation angered the Americans and led in part to a U.S. declaration of war in April

Technical age: from WWI to Shannon's advent

The **Zimmermann** telegram



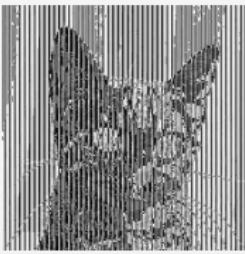
- ▶ It was a 1917 diplomatic proposal from the **German Empire** to **Mexico** to make war against the **United States**
- ▶ The proposal was **caught** by the **British** before it could get to Mexico
- ▶ It was intercepted and decoded by the **British** cryptographers of **Room 40**
- ▶ The revelation angered the Americans and led in part to a **U.S. declaration of war in April**

Technical age: from WWI to Shannon's advent

XOR

XOR or the addition operation $(\text{mod } 2)$ on bits $(0, 1)$

XOR		
\oplus	0	1
0	0	1
1	1	0



Longueur de clef = 4
Longueur de clef = 10

Longueur de clef = 40
Longueur de clef = 100

Longueur de clef = 400
Longueur de clef = 4000

Technical age: from WWI to Shannon's advent

XOR

XOR or the addition operation $(\text{mod } 2)$ on bits (0, 1)

XOR		
\oplus	0	1
0	0	1
1	1	0



Longueur de clef = 10

Longueur de clef = 40
Longueur de clef = 100

Longueur de clef = 400
Longueur de clef = 4000

Technical age: from WWI to Shannon's advent

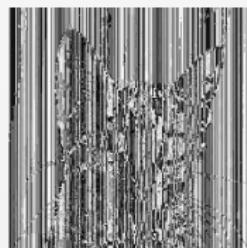
XOR

XOR or the addition operation $(\text{mod } 2)$ on bits (0, 1)

XOR		
\oplus	0	1
0	0	1
1	1	0



Longueur de clef = 4



Longueur de clef = 100

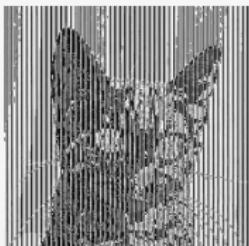
Longueur de clef = 400
Longueur de clef = 4000

Technical age: from WWI to Shannon's advent

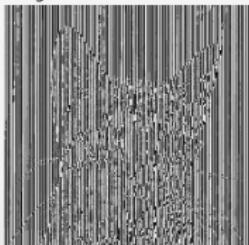
XOR

XOR or the addition operation $(\text{mod } 2)$ on bits (0, 1)

XOR		
\oplus	0	1
0	0	1
1	1	0



Longueur de clef = 4



Longueur de clef = 10



Longueur de clef = 40



Longueur de clef = 100

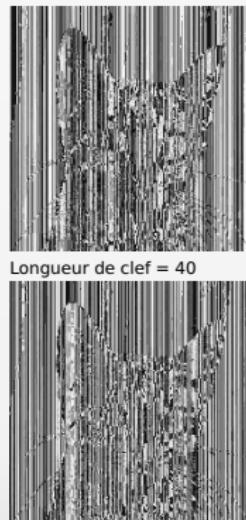
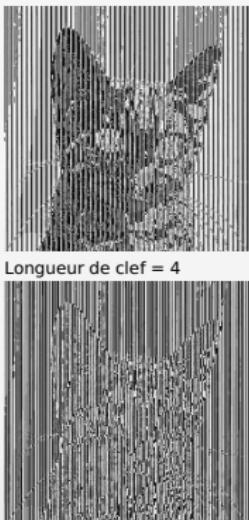
Longueur de clef = 400
Longueur de clef = 4000

Technical age: from WWI to Shannon's advent

XOR

XOR or the addition operation $(\text{mod } 2)$ on bits (0, 1)

XOR		
\oplus	0	1
0	0	1
1	1	0



Technical age: from WWI to Shannon's advent

XOR

XOR or the addition operation $(\text{mod } 2)$ on bits (0, 1)

XOR		
\oplus	0	1
0	0	1
1	1	0



Longueur de clef = 4



Longueur de clef = 40



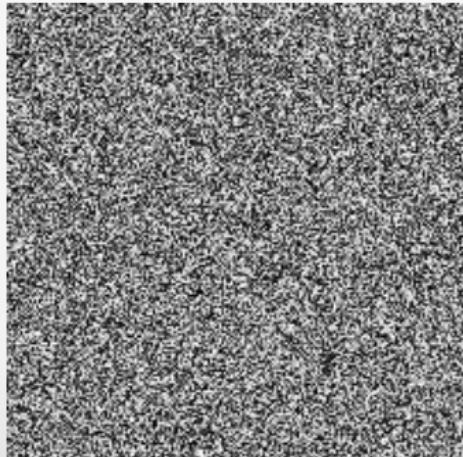
Longueur de clef = 400

Longueur de clef = 4000

Technical age: from WWI to Shannon's advent

The only provably secure cipher

In 1917, **Gilbert Vernam**, an American Telephone and Telegraph Company (AT&T) engineer, created a machine that makes a non-repeating, virtually random sequence of characters which is called a **one-time pad**.



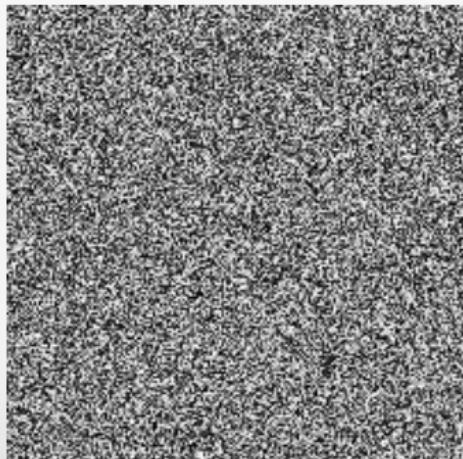
- ▶ Ciphering $C = M \oplus k$
- ▶ Deciphering $M = C \oplus k$
- ▶ The encryption key and the message have the same length
- ▶ By never using twice a key the Vernam cipher is the only proven method of securely communicating
- ▶ This cipher algorithm was first used to

red phone

Technical age: from WWI to Shannon's advent

The only provably secure cipher

In 1917, **Gilbert Vernam**, an American Telephone and Telegraph Company (AT&T) engineer, created a machine that makes a non-repeating, virtually random sequence of characters which is called a **one-time pad**.



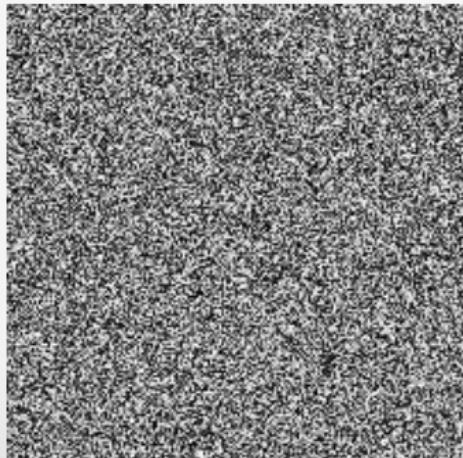
- ▶ Ciphering $C = M \oplus k$
- ▶ Deciphering $M = C \oplus k$
- ▶ The encryption key and the message have **the same length**
- ▶ By **never using twice** a key the Vernam cipher is **the only proven method** of securely communicating
- ▶ This cipher algorithm was first used to

red phone

Technical age: from WWI to Shannon's advent

The only provably secure cipher

In 1917, **Gilbert Vernam**, an American Telephone and Telegraph Company (AT&T) engineer, created a machine that makes a non-repeating, virtually random sequence of characters which is called a **one-time pad**.

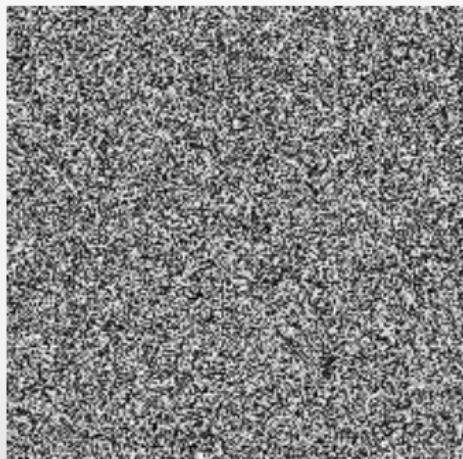


- ▶ Ciphering $C = M \oplus k$
- ▶ Deciphering $M = C \oplus k$
- ▶ The encryption key and the message have **the same length**
- ▶ By **never using twice** a key the Vernam cipher is **the only proven method** of securely communicating
- ▶ This cipher algorithm was first used to protect communications through the **telephone**

Technical age: from WWI to Shannon's advent

The only provably secure cipher

In 1917, **Gilbert Vernam**, an American Telephone and Telegraph Company (AT&T) engineer, created a machine that makes a non-repeating, virtually random sequence of characters which is called a **one-time pad**.

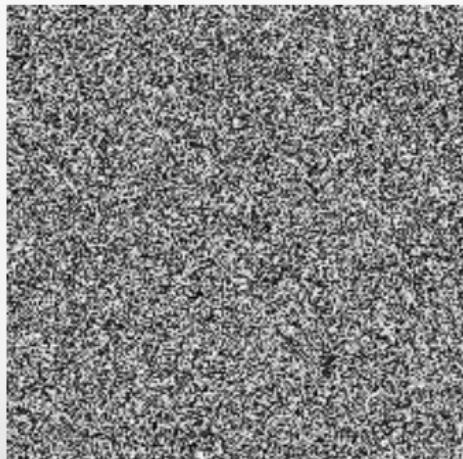


- ▶ Ciphering $C = M \oplus k$
- ▶ Deciphering $M = C \oplus k$
- ▶ The encryption key and the message have **the same length**
- ▶ By **never using twice** a key the Vernam cipher is **the only proven method** of securely communicating
- ▶ This cipher algorithm was first used to protect communications through the **red phone**

Technical age: from WWI to Shannon's advent

The only provably secure cipher

In 1917, **Gilbert Vernam**, an American Telephone and Telegraph Company (AT&T) engineer, created a machine that makes a non-repeating, virtually random sequence of characters which is called a **one-time pad**.

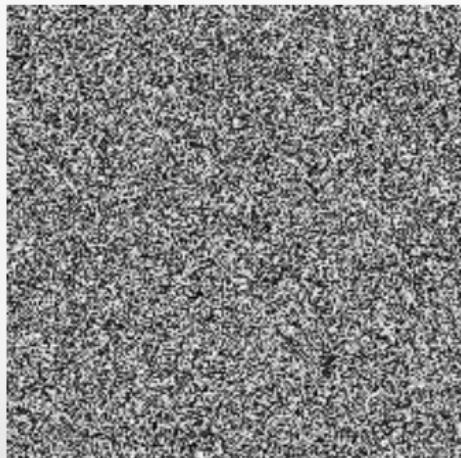


- ▶ Ciphering $C = M \oplus k$
- ▶ Deciphering $M = C \oplus k$
- ▶ The encryption key and the message have **the same length**
- ▶ By **never using twice** a key the Vernam cipher is **the only proven method** of securely communicating
- ▶ This cipher algorithm was first used to protect communications through the **red phone**

Technical age: from WWI to Shannon's advent

The only provably secure cipher

In 1917, **Gilbert Vernam**, an American Telephone and Telegraph Company (AT&T) engineer, created a machine that makes a non-repeating, virtually random sequence of characters which is called a **one-time pad**.



- ▶ Ciphering $C = M \oplus k$
- ▶ Deciphering $M = C \oplus k$
- ▶ The encryption key and the message have **the same length**
- ▶ By **never using twice** a key the Vernam cipher is **the only proven method** of securely communicating
- ▶ This cipher algorithm was first used to protect communications through the **red phone**

Technical age: from WWI to Shannon's advent

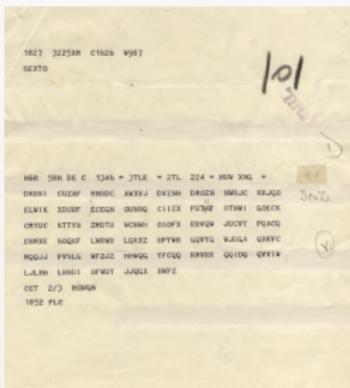
The Enigma



- ▶ developed and patented by a German Arthur Scherbius in 1919
- ▶ it uses rotors for multi substitution
- ▶ plug switchboard
- ▶ 159×10^{18} possibles keys

Technical age: from WWI to Shannon's advent

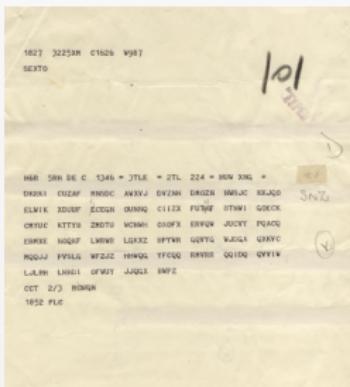
The Enigma



- ▶ developed and patented by a German Arthur Scherbius in 1919
- ▶ it uses rotors for multi substitution
- ▶ plug switchboard
- ▶ 159×10^{18} possibles keys

Technical age: from WWI to Shannon's advent

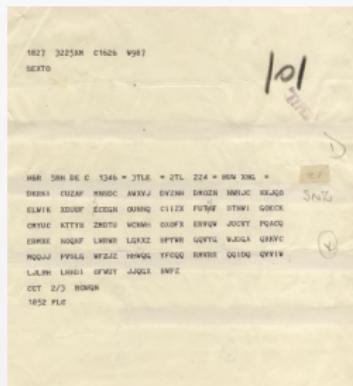
The Enigma



- ▶ developed and patented by a German Arthur Scherbius in 1919
- ▶ it uses rotors for multi substitution
- ▶ plug switchboard
- ▶ 159×10^{18} possibles keys

Technical age: from WWI to Shannon's advent

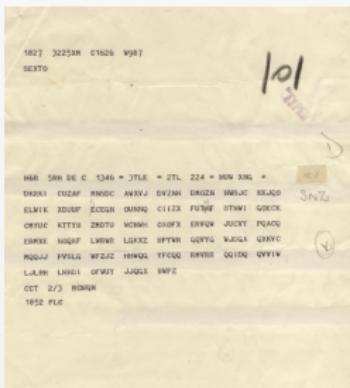
The Enigma



- ▶ developed and patented by a German Arthur Scherbius in 1919
- ▶ it uses rotors for multi substitution
- ▶ plug switchboard
- ▶ 159×10^{18} possibles keys

Technical age: from WWI to Shannon's advent

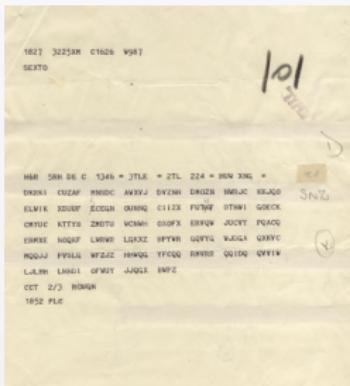
The Enigma



- ▶ developed and patented by a German Arthur Scherbius in 1919
- ▶ it uses rotors for multi substitution
- ▶ plug switchboard
- ▶ 159×10^{18} possibles keys

Technical age: from WWI to Shannon's advent

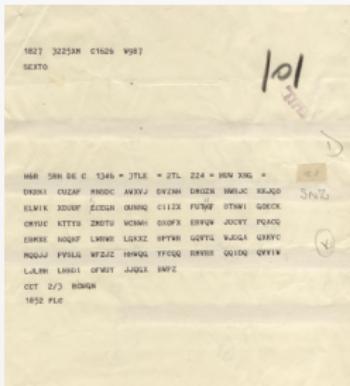
The Enigma



- ▶ developed and patented by a German Arthur Scherbius in 1919
- ▶ it uses rotors for multi substitution
- ▶ plug switchboard
- ▶ 159×10^{18} possibles keys

Technical age: from WWI to Shannon's advent

The Enigma



- ▶ developed and patented by a German Arthur Scherbius in 1919
- ▶ it uses rotors for multi substitution
- ▶ plug switchboard
- ▶ 159×10^{18} possibles keys



Technical age: from WWI to Shannon's advent

The Navajo "Code Talkers"

- ▶ During the **World War II** Americans, by taking advantage of the complexity of their language, employed **Navajo Indians** as radio operators
- ▶ The Navajo operator translated the message into **Navajo dialect** and then transmitted by radio. At the reception, another operator Navajo, translated the message in English to pass it on to his superiors
- ▶ **"The Japanese were never able to break the "code"**



Technical age: from WWI to Shannon's advent

The Navajo "Code Talkers"

- ▶ During the **World War II** Americans, by taking advantage of the complexity of their language, employed **Navajo Indians** as radio operators
- ▶ The Navajo operator translated the message into **Navajo dialect** and then transmitted by radio. At the reception, another operator Navajo, translated the message in English to pass it on to his superiors
- ▶ The Japanese were never able to break the "code"



Technical age: from WWI to Shannon's advent

The Navajo "Code Talkers"

- ▶ During the **World War II** Americans, by taking advantage of the complexity of their language, employed **Navajo Indians** as radio operators
- ▶ The Navajo operator translated the message into **Navajo dialect** and then transmitted by radio. At the reception, another operator Navajo, translated the message in English to pass it on to his superiors
- ▶ The Japanese were never able to break the "code"



Technical age: from WWI to Shannon's advent

The Navajo "Code Talkers"

- ▶ During the **World War II** Americans, by taking advantage of the complexity of their language, employed **Navajo Indians** as radio operators
- ▶ The Navajo operator translated the message into **Navajo dialect** and then transmitted by radio. At the reception, another operator Navajo, translated the message in English to pass it on to his superiors
- ▶ **The Japanese were never able to break the "code"**

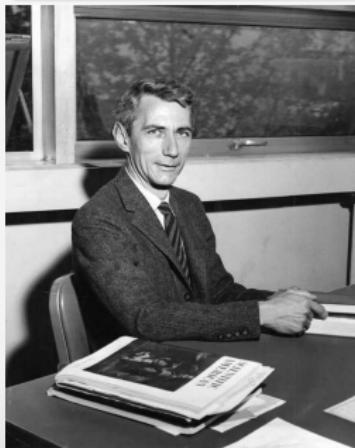


Technical age: from WWI to Shannon's advent

Conclusion: technical age

"Breaking a good cipher should require as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type."

Shannon



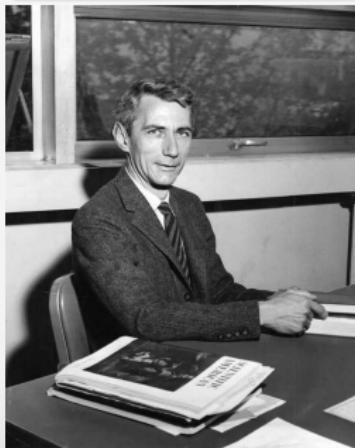
- ➊ The cryptanalysts have yet the advantage
- ➋ The Cryptography leaves the army's world to enter to the mathematicians world
- ➌ Emergence of the communication theory

Technical age: from WWI to Shannon's advent

Conclusion: technical age

"Breaking a good cipher should require as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type."

Shannon



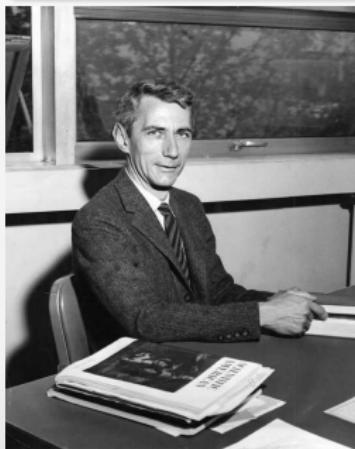
- ➊ The cryptanalysts have yet the advantage
- ➋ The Cryptography leaves the army's world to enter to the mathematicians world
- ➌ Emergence of the communication theory

Technical age: from WWI to Shannon's advent

Conclusion: technical age

"Breaking a good cipher should require as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type."

Shannon



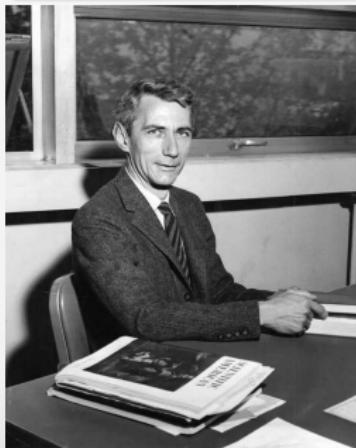
- ① The cryptanalysts have yet the advantage
- ② The Cryptography leaves the army's world to enter to the mathematicians world
- ③ Emergence of the communication theory

Technical age: from WWI to Shannon's advent

Conclusion: technical age

"Breaking a good cipher should require as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type."

Shannon



- ① The cryptanalysts have yet the advantage
- ② The Cryptography leaves the army's world to enter to the mathematicians world
- ③ Emergence of the communication theory

Third period – modern age: from the end of WW II to our days

3 History

- First period – craftsmanship age: from the origins to WW I
- Second period – technical age: from WWI to Shannon's advent
- Third period – modern age: from the end of WW II to our days



Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES:** 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES:** 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA:** 128 bits of key length and a data block size of 64 bits
- ▶ **AES:** 128, 192 or 256 bits of key length and a data block size of 128 bits

Stream cipher

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES:** 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES:** 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA:** 128 bits of key length and a data block size of 64 bits
- ▶ **AES:** 128, 196 or 256 bits of key length and a data block size of 128 bits

Stream cipher

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES:** 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES:** 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA:** 128 bits of key length and a data block size of 64 bits
- ▶ **AES:** 128, 196 or 256 bits of key length and a data block size of 128 bits

Stream cipher

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES**: 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES**: 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA**: 128 bits of key length and a data block size of 64 bits
- ▶ **AES**: 128, 196 or 256 bits of key length and a data block size of 128 bits

Stream cipher

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES**: 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES**: 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA**: 128 bits of key length and a data block size of 64 bits
- ▶ **AES**: 128, 196 or 256 bits of key length and a data block size of 128 bits

Stream cipher

→ stream cipher → stream generator

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES**: 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES**: 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA**: 128 bits of key length and a data block size of 64 bits
- ▶ **AES**: 128, 196 or 256 bits of key length and a data block size of 128 bits

Stream cipher

- ▶ **Vernam**: one time pad generator
- ▶ **RC4**: 40 to 128 bits of key length, *used in SSL or WEP*
- ▶ **E0**: 128 bits of key length, *used by bluetooth*
- ▶ **A5/1**: 64 bits of key length, *used by GSM*

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES**: 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES**: 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA**: 128 bits of key length and a data block size of 64 bits
- ▶ **AES**: 128, 196 or 256 bits of key length and a data block size of 128 bits

Stream cipher

- ▶ **Vernam**: one time pad generator
- ▶ **RC4**: 40 to 128 bits of key length, *used in SSL or WEP*
- ▶ **E0**: 128 bits of key length, *used by bluetooth*
- ▶ **A5/1**: 64 bits of key length, *used by GSM*

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES**: 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES**: 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA**: 128 bits of key length and a data block size of 64 bits
- ▶ **AES**: 128, 196 or 256 bits of key length and a data block size of 128 bits

Stream cipher

- ▶ **Vernam**: one time pad generator
- ▶ **RC4**: 40 to 128 bits of key length, *used in SSL or WEP*
- ▶ **E0**: 128 bits of key length, *used by bluetooth*
- ▶ **A5/1**: 64 bits of key length, *used by GSM*

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES**: 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES**: 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA**: 128 bits of key length and a data block size of 64 bits
- ▶ **AES**: 128, 196 or 256 bits of key length and a data block size of 128 bits

Stream cipher

- ▶ **Vernam**: one time pad generator
- ▶ **RC4**: 40 to 128 bits of key length, *used in SSL or WEP*
- ▶ **E0**: 128 bits of key length, *used by bluetooth*
- ▶ **A5/1**: 64 bits of key length, *used by GSM*

Modern age: from the end of WW II to our days

Development of **symmetric** encryption algorithms

Block cipher

- ▶ **DES**: 56 bits of key length and a data block size of 64 bits
- ▶ **triple DES**: 112 bits of key length and a data block size of 168 bits
- ▶ **IDEA**: 128 bits of key length and a data block size of 64 bits
- ▶ **AES**: 128, 196 or 256 bits of key length and a data block size of 128 bits

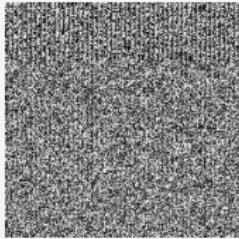
Stream cipher

- ▶ **Vernam**: one time pad generator
- ▶ **RC4**: 40 to 128 bits of key length, *used in SSL or WEP*
- ▶ **E0**: 128 bits of key length, *used by bluetooth*
- ▶ **A5/1**: 64 bits of key length, *used by GSM*

Modern age: from the end of WW II to our days

Composition of cipher

Product by composition of two cryptographic systems: $C_k(x) = C'_{k_1}(C''_{k_2}(x))$



Permutation and Substitution (4)
Permutation and Substitution (10)

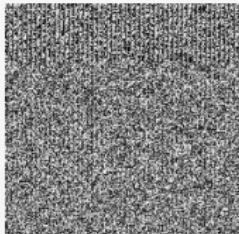
Permutation and Substitution (40)
Substitution and Permutation (4)

Substitution and Permutation (10)
Substitution and Permutation (40)

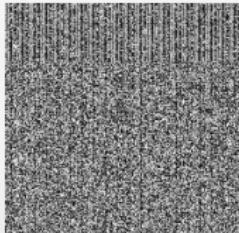
Modern age: from the end of WW II to our days

Composition of cipher

Product by composition of two cryptographic systems: $C_k(x) = C'_{k_1}(C''_{k_2}(x))$



Permutation and Substitution (4)



Permutation and Substitution (10)

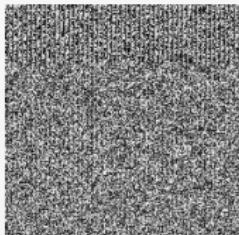
Permutation and Substitution (40)
Substitution and Permutation (4)

Substitution and Permutation (10)
Substitution and Permutation (40)

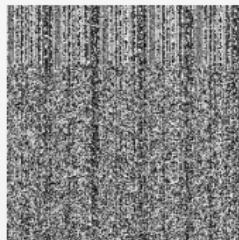
Modern age: from the end of WW II to our days

Composition of cipher

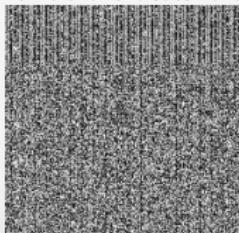
Product by composition of two cryptographic systems: $C_k(x) = C'_{k_1}(C''_{k_2}(x))$



Permutation and Substitution (4)



Permutation and Substitution (40)
Substitution and Permutation (4)



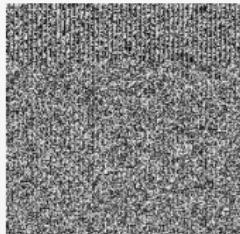
Permutation and Substitution (10)

Substitution and Permutation (10)
Substitution and Permutation (40)

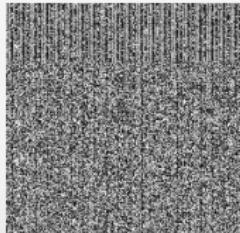
Modern age: from the end of WW II to our days

Composition of cipher

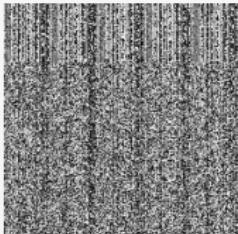
Product by composition of two cryptographic systems: $C_k(x) = C'_{k_1}(C''_{k_2}(x))$



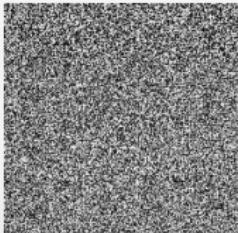
Permutation and Substitution (4)



Permutation and Substitution (10)



Permutation and Substitution (40)



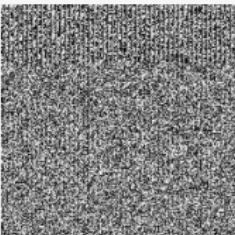
Substitution and Permutation (4)

Substitution and Permutation (10)
Substitution and Permutation (40)

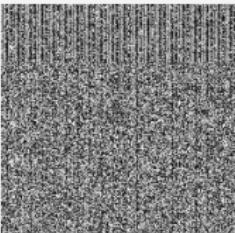
Modern age: from the end of WW II to our days

Composition of cipher

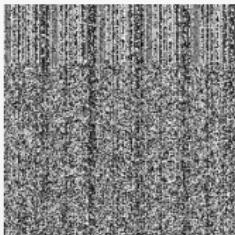
Product by composition of two cryptographic systems: $C_k(x) = C'_{k_1}(C''_{k_2}(x))$



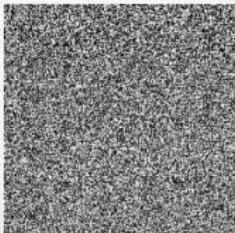
Permutation and Substitution (4)



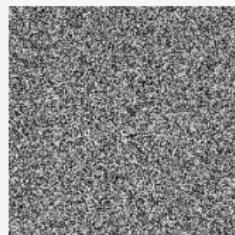
Permutation and Substitution (10)



Permutation and Substitution (40)



Substitution and Permutation (4)

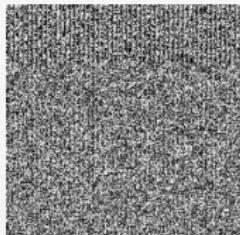


Substitution and Permutation (10)
Substitution and Permutation (40)

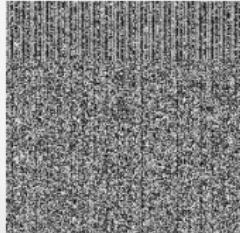
Modern age: from the end of WW II to our days

Composition of cipher

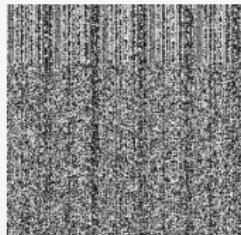
Product by composition of two cryptographic systems: $C_k(x) = C'_{k_1}(C''_{k_2}(x))$



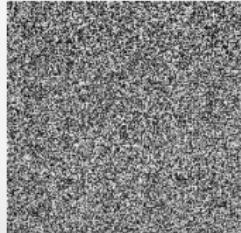
Permutation and Substitution (4)



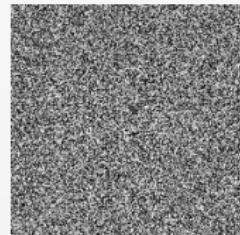
Permutation and Substitution (10)



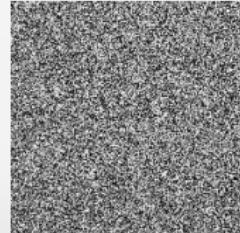
Permutation and Substitution (40)



Substitution and Permutation (4)



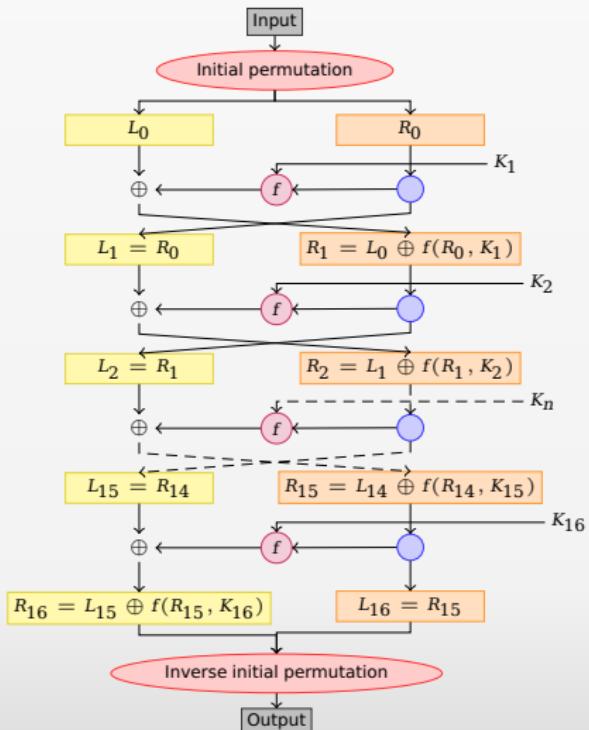
Substitution and Permutation (10)



Substitution and Permutation (40)

Modern age: from the end of WW II to our days

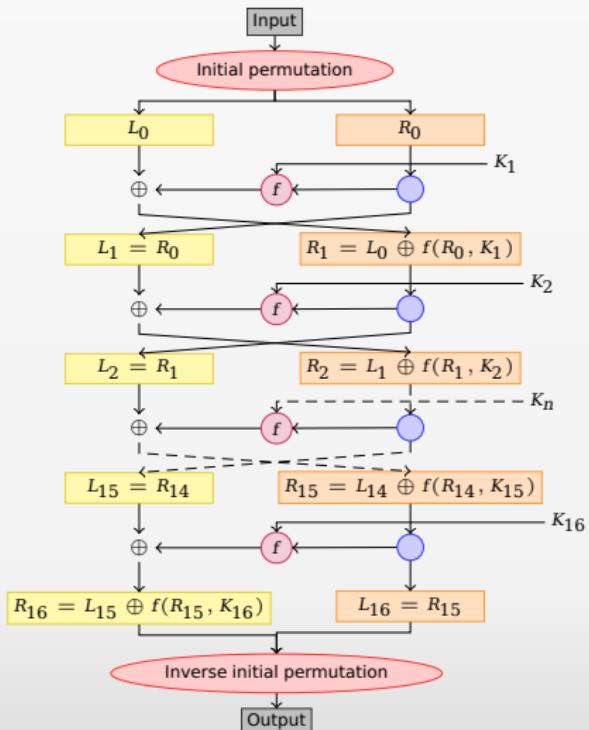
The Data Encryption Standard – DES



- ▶ DES is a standard published by the NIST on 1977 under the FIPS 46 number
- ▶ It is the inheritor of the encryption algorithm LUCIFER published by IBM
- ▶ The block size is 64 bits and the key size is 56 bits
- ▶ 16 rounds with an initial and final permutation, termed IP and FP, which are inverses
- ▶ Before the main rounds, the block is divided into two 32 bits halves and
- ▶ network.)

Modern age: from the end of WW II to our days

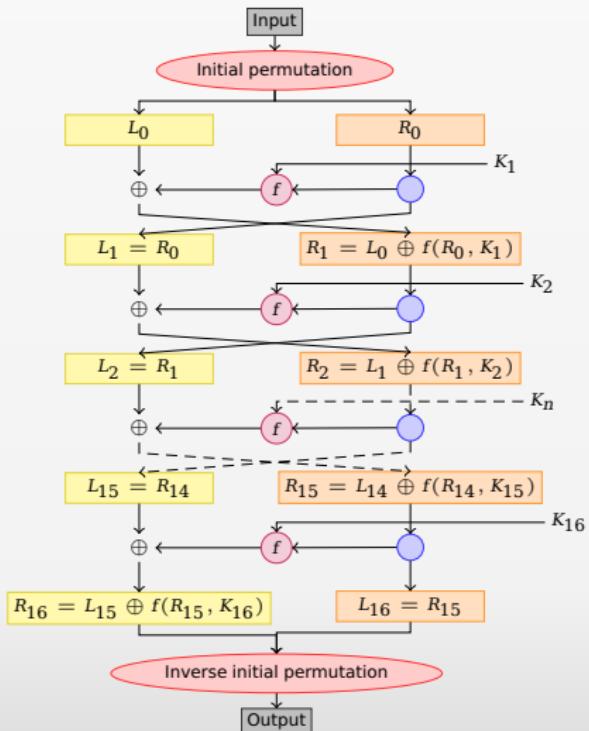
The Data Encryption Standard – DES



- ▶ DES is a standard published by the NIST on 1977 under the **FIPS 46** number
- ▶ It is the inheritor of the encryption algorithm **LUCIFER** published by IBM
- ▶ The block size is **64 bits** and the key size is **56 bits**
- ▶ **16 rounds** with an initial and final permutation, termed **IP** and **FP**, which are inverses
- ▶ Before the main rounds, the block is divided into **two 32 bits halves** and processed alternately (Feistel)

Modern age: from the end of WW II to our days

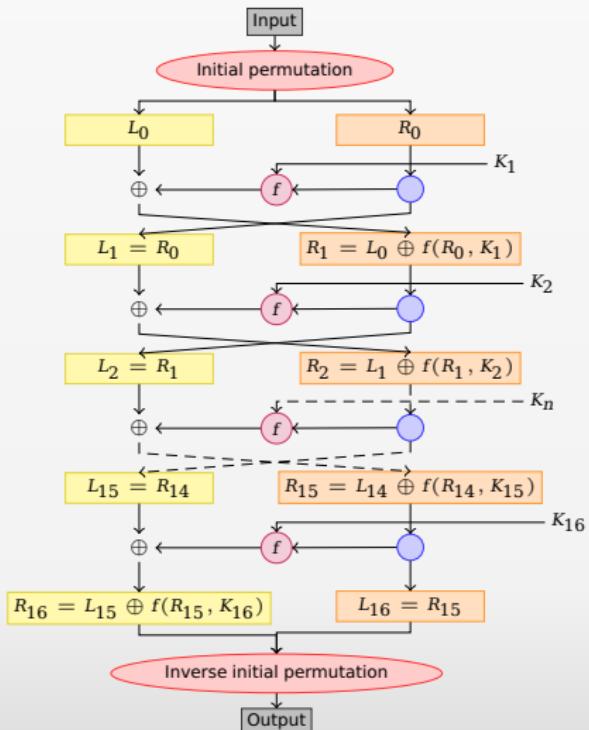
The Data Encryption Standard – DES



- ▶ DES is a standard published by the NIST on 1977 under the **FIPS 46** number
- ▶ It is the inheritor of the encryption algorithm **LUCIFER** published by IBM
- ▶ The block size is **64 bits** and the key size is **56 bits**
- ▶ 16 rounds with an initial and final permutation, termed **IP** and **FP**, which are inverses
- ▶ Before the main rounds, the block is divided into two 32 bits halves and processed alternately (Feistel network)

Modern age: from the end of WW II to our days

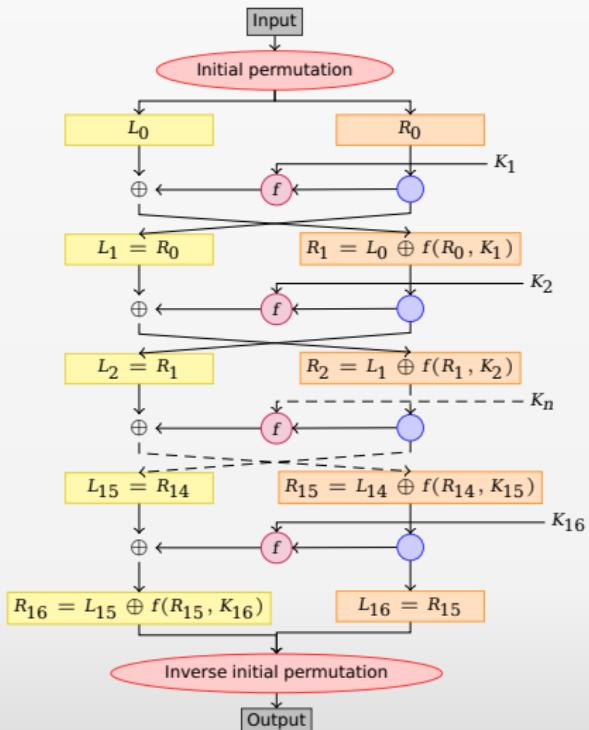
The Data Encryption Standard – DES



- ▶ DES is a standard published by the NIST on 1977 under the **FIPS 46** number
- ▶ It is the inheritor of the encryption algorithm **LUCIFER** published by IBM
- ▶ The block size is **64 bits** and the key size is **56 bits**
- ▶ **16 rounds** with an initial and final permutation, termed **IP** and **FP**, which are inverses
- ▶ Before the main rounds, the block is divided into **two 32 bits halves** and processed alternately (Feistel network)

Modern age: from the end of WW II to our days

The Data Encryption Standard – DES



- ▶ DES is a standard published by the NIST on 1977 under the **FIPS 46** number
- ▶ It is the inheritor of the encryption algorithm **LUCIFER** published by IBM
- ▶ The block size is **64 bits** and the key size is **56 bits**
- ▶ **16 rounds** with an initial and final permutation, termed **IP** and **FP**, which are inverses
- ▶ Before the main rounds, the block is divided into **two 32 bits halves** and processed alternately (Feistel network)

Modern age: from the end of WW II to our days

The Data Encryption Standard – DES

Practice with Sage: Simplified DES

```
# import library
from sage.crypto.block_cipher.sdes import SimplifiedDES
from sage.crypto.util import bin_to_ascii

# initialize cipher
sdes = SimplifiedDES(); sdes
bin = BinaryStrings(); bin
K = sdes.list_to_string(sdes.random_key()); K
P = bin.encoding("Encrypt this using S-DES!"); P

# for simplified DES the binary string must be positive and a multiple of 8
Mod(len(P), 8) == 0

# ciphering
C = sdes(P, K, algorithm="encrypt"); C
# deciphering
DC = sdes(C, K, algorithm="decrypt"); DC

# control result
DC == P
msg = DC = bin_to_ascii(plaintext); msg
```

Modern age: from the end of WW II to our days

The Advanced Encryption Standard – AES

- ▶ AES is a standard published by the NIST on 2001 under the FIPS 197 number
- ▶ It is the successor of the DES standard
- ▶ It is the inheritor of the encryption algorithm RIJNDAEL developed by Joan Daemen and Vincent Rijmen
- ▶ The block size is 128 bits and the key size is 128, 192 or 256 bits
- ▶ 10, 12, 14 rounds with an initial and final permutation

Modern age: from the end of WW II to our days

The Advanced Encryption Standard – AES

- ▶ AES is a standard published by the **NIST** on 2001 under the **FIPS 197** number
- ▶ It is the successor of the DES standard
- ▶ It is the inheritor of the encryption algorithm **RIJNDAEL** developed by **Joan Daemen** and **Vincent Rijmen**
- ▶ The block size is **128 bits** and the key size is **128, 192 or 256 bits**
- ▶ **10, 12, 14 rounds** with an initial and final permutation

Modern age: from the end of WW II to our days

The Advanced Encryption Standard – AES

- ▶ AES is a standard published by the **NIST** on 2001 under the **FIPS 197** number
- ▶ It is the successor of the DES standard
- ▶ It is the inheritor of the encryption algorithm **RIJNDAEL** developed by **Joan Daemen** and **Vincent Rijmen**
- ▶ The block size is **128 bits** and the key size is **128, 192 or 256 bits**
- ▶ **10, 12, 14 rounds** with an initial and final permutation

Modern age: from the end of WW II to our days

The Advanced Encryption Standard – AES

- ▶ AES is a standard published by the **NIST** on 2001 under the **FIPS 197** number
- ▶ It is the successor of the DES standard
- ▶ It is the inheritor of the encryption algorithm **RIJNDAEL** developed by **Joan Daemen** and **Vincent Rijmen**
- ▶ The block size is **128 bits** and the key size is **128, 192 or 256 bits**
- ▶ **10, 12, 14 rounds** with an initial and final permutation

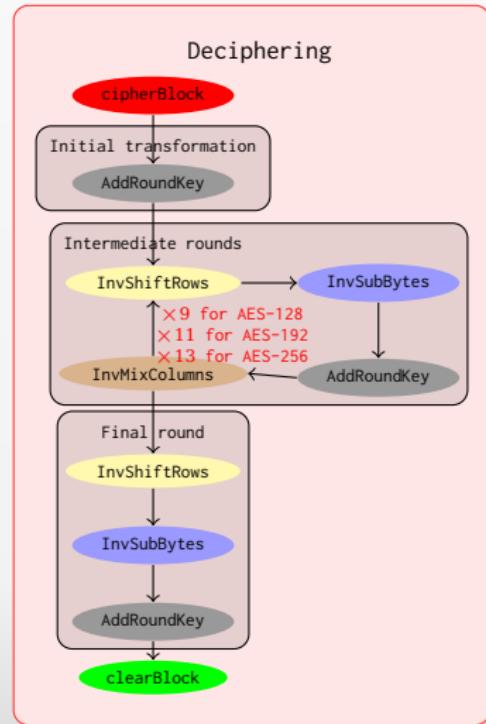
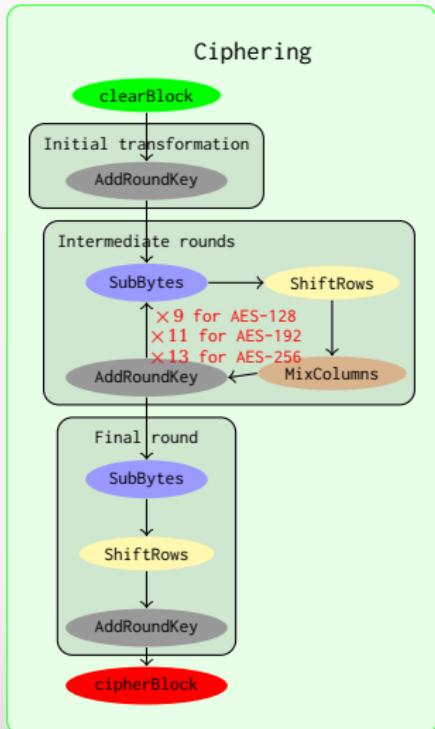
Modern age: from the end of WW II to our days

The Advanced Encryption Standard – AES

- ▶ AES is a standard published by the **NIST** on 2001 under the **FIPS 197** number
- ▶ It is the successor of the DES standard
- ▶ It is the inheritor of the encryption algorithm **RIJNDAEL** developed by **Joan Daemen** and **Vincent Rijmen**
- ▶ The block size is **128 bits** and the key size is **128, 192 or 256 bits**
- ▶ **10, 12, 14 rounds** with an initial and final permutation

Modern age: from the end of WW II to our days

The Advanced Encryption Standard – AES



Modern age: from the end of WW II to our days

The Advanced Encryption Standard – AES

Practice with Sage: mini-AES

```
# import library
from sage.crypto.block_cipher.miniaes import MiniAES
from sage.crypto.util import bin_to_ascii

# initialize cipher
maes = MiniAES(); maes
bin = BinaryStrings(); bin
key = bin.encoding("KE"); key
P = bin.encoding("Encrypt this secret message!"); P

# ciphering
C = maes(P, key, algorithm="encrypt"); C
# deciphering
DC = maes(C, key, algorithm="decrypt"); DC

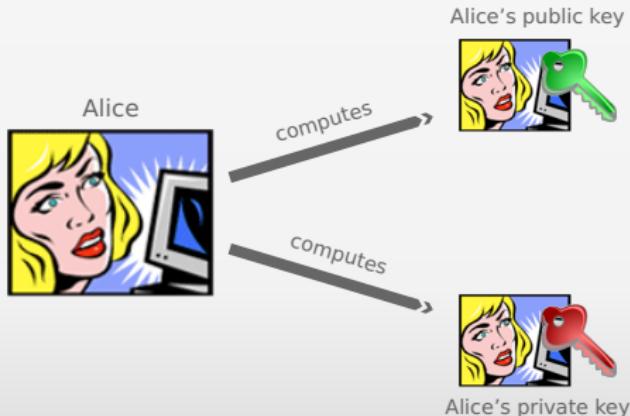
# control result
DC == P
msg = bin_to_ascii(DC); msg
```

Modern age: from the end of WW II to our days

Public-key cryptography

- ▶ **Asymmetric** cryptography or **Public-key** cryptography
- ▶ Based on the existence of **one-way trapdoor** functions
- ▶ Usages:

• Key exchange, digital signatures, non-repudiation, secure communication

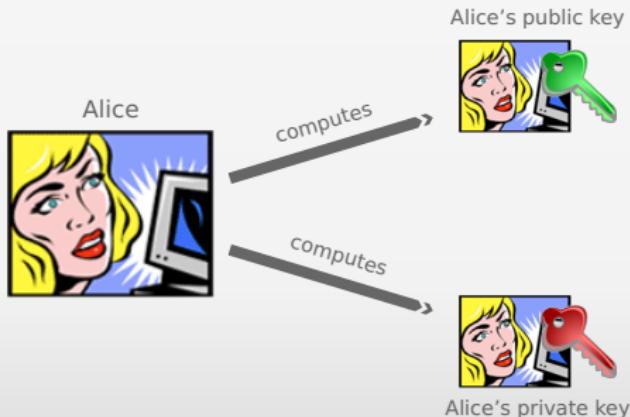


Modern age: from the end of WW II to our days

Public-key cryptography

- ▶ **Asymmetric** cryptography or **Public-key** cryptography
- ▶ Based on the existence of **one-way trapdoor** functions
- ▶ Usages:

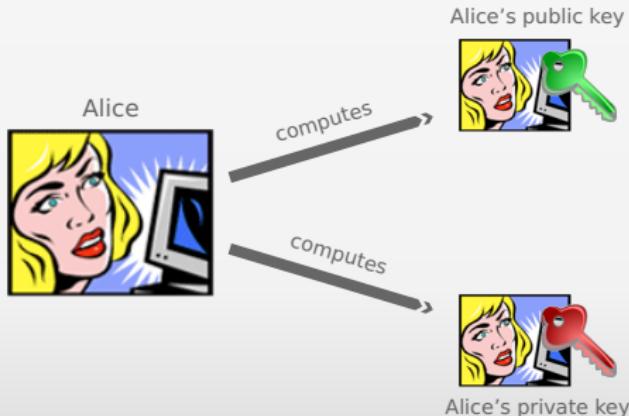
key-exchange algorithm – Diffie-Hellman key exchange



Modern age: from the end of WW II to our days

Public-key cryptography

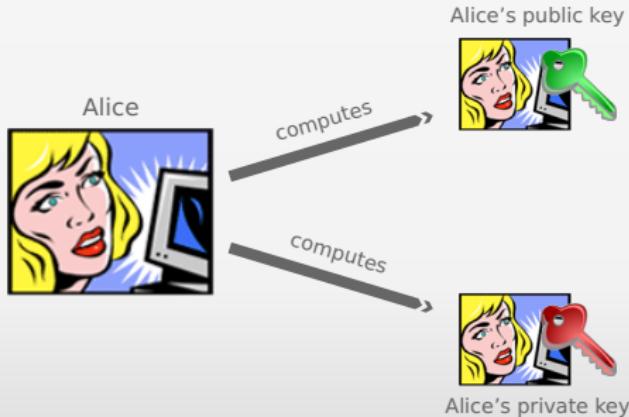
- ▶ **Asymmetric** cryptography or **Public-key** cryptography
- ▶ Based on the existence of **one-way trapdoor functions**
- ▶ Usages:
 - ⇒ key-exchange algorithm – Diffie–Hellman key exchange
 - ⇒ Public key encryption – RSA



Modern age: from the end of WW II to our days

Public-key cryptography

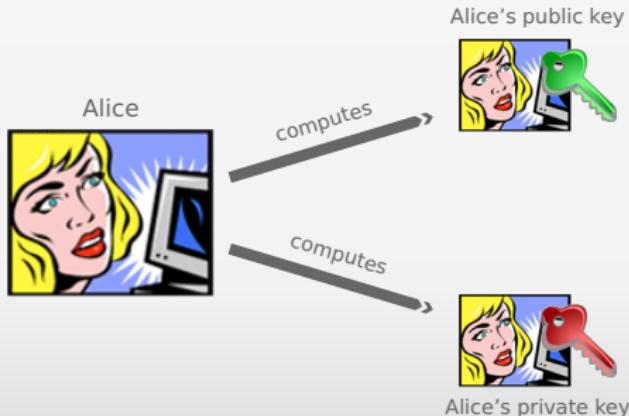
- ▶ **Asymmetric** cryptography or **Public-key** cryptography
- ▶ Based on the existence of **one-way trapdoor functions**
- ▶ Usages:
 - ▶ key-exchange algorithm – Diffie–Hellman key exchange
 - ▶ Public key encryption – RSA
 - ▶ Digital signatures – RSA



Modern age: from the end of WW II to our days

Public-key cryptography

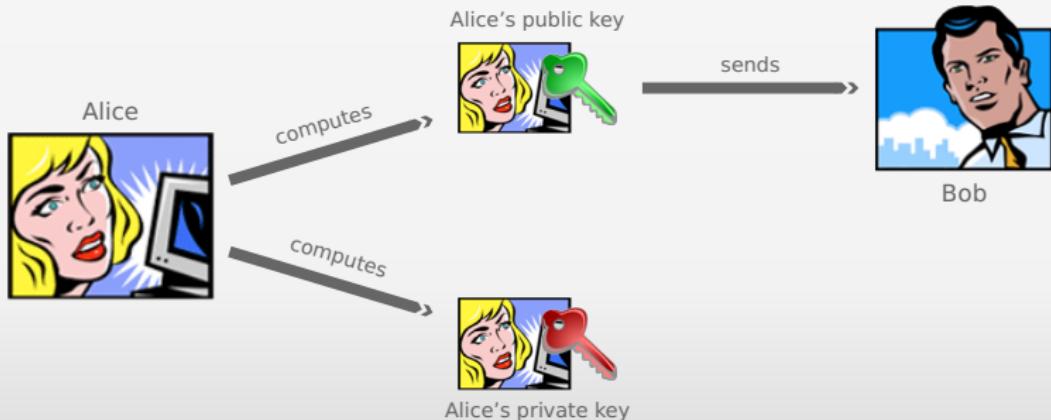
- ▶ **Asymmetric** cryptography or **Public-key** cryptography
- ▶ Based on the existence of **one-way trapdoor functions**
- ▶ Usages:
 - ▶ key-exchange algorithm – Diffie–Hellman key exchange
 - ▶ Public key encryption – RSA
 - ▶ Digital signatures – RSA



Modern age: from the end of WW II to our days

Public-key cryptography

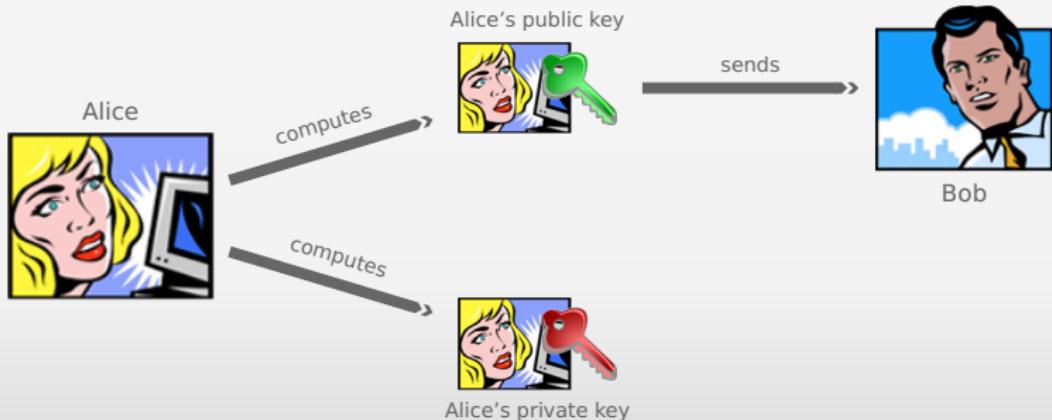
- ▶ **Asymmetric** cryptography or **Public-key** cryptography
- ▶ Based on the existence of **one-way trapdoor functions**
- ▶ Usages:
 - ▶ key-exchange algorithm – Diffie–Hellman key exchange
 - ▶ Public key encryption – RSA
 - ▶ Digital signatures – RSA



Modern age: from the end of WW II to our days

Public-key cryptography

- ▶ **Asymmetric** cryptography or **Public-key** cryptography
- ▶ Based on the existence of **one-way trapdoor functions**
- ▶ Usages:
 - ▶ key-exchange algorithm – Diffie–Hellman key exchange
 - ▶ Public key encryption – RSA
 - ▶ Digital signatures – RSA



Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

- ▶ The **Diffie–Hellman** key agreement was invented in **1976** during a collaboration between **Whitfield Diffie** and **Martin Hellman**
- ▶ This protocol allows two parties that have no prior knowledge of each other to jointly **establish a shared secret key** over an insecure communications channel
- ▶ Based on the **discrete logarithm problem** which is to find x given g^x

Network Working Group
Request for Comments: 2631
Category: Standards Track

E. Rescorla
RTFM Inc.
June 1999

Diffie-Hellman Key Agreement Method

...

Abstract

This document standardizes one particular Diffie-Hellman variant, based on the ANSI X9.42 draft, developed by the ANSI X9F1 working group. Diffie-Hellman is a key agreement algorithm used by two parties to agree on a shared secret. An algorithm for converting the shared secret into an arbitrary amount of keying material is provided. The resulting keying material is used as a symmetric encryption key. The Diffie-Hellman variant described requires the recipient to have a certificate, but the originator may have a static key pair (with the public key placed in a certificate) or an ephemeral key pair.

Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

- ▶ The **Diffie–Hellman** key agreement was invented in **1976** during a collaboration between **Whitfield Diffie** and **Martin Hellman**
- ▶ This protocol allows two parties that have no prior knowledge of each other to jointly **establish a shared secret key** over an insecure communications channel
- ▶ Based on the **discrete logarithm problem** which is to find x given g^x

Network Working Group
Request for Comments: 2631
Category: Standards Track

E. Rescorla
RTFM Inc.
June 1999

Diffie-Hellman Key Agreement Method

...

Abstract

This document standardizes one particular Diffie-Hellman variant, based on the ANSI X9.42 draft, developed by the ANSI X9F1 working group. Diffie-Hellman is a key agreement algorithm used by two parties to agree on a shared secret. An algorithm for converting the shared secret into an arbitrary amount of keying material is provided. The resulting keying material is used as a symmetric encryption key. The Diffie-Hellman variant described requires the recipient to have a certificate, but the originator may have a static key pair (with the public key placed in a certificate) or an ephemeral key pair.

Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

- ▶ The **Diffie–Hellman** key agreement was invented in **1976** during a collaboration between **Whitfield Diffie** and **Martin Hellman**
- ▶ This protocol allows two parties that have no prior knowledge of each other to jointly **establish a shared secret key** over an insecure communications channel
- ▶ Based on the **discrete logarithm problem** which is to find x given g^x

Network Working Group
Request for Comments: 2631
Category: Standards Track

E. Rescorla
RTFM Inc.
June 1999

Diffie-Hellman Key Agreement Method

...

Abstract

This document standardizes one particular Diffie-Hellman variant, based on the ANSI X9.42 draft, developed by the ANSI X9F1 working group. Diffie-Hellman is a key agreement algorithm used by two parties to agree on a shared secret. An algorithm for converting the shared secret into an arbitrary amount of keying material is provided. The resulting keying material is used as a symmetric encryption key. The Diffie-Hellman variant described requires the recipient to have a certificate, but the originator may have a static key pair (with the public key placed in a certificate) or an ephemeral key pair.

Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

Diffie – Hellman: operations

Alice

Eve

Bob

Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

Diffie – Hellman: operations

Step 1

Alice

Alice and Bob exchange a Prime P and a Base B in clear text, such that $P > B$ and B is primitive root of P .

$$P = 23, B = 5$$

Eve

Eve sees
 $P = 23, B = 5$

Bob

Alice and Bob exchange a Prime P and a Base B in clear text, such that $P > B$ and B is primitive root of P .

$$P = 23, B = 5$$

Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

Diffie – Hellman: operations

	Alice	Eve	Bob
Step 1	Alice and Bob exchange a Prime P and a Base B in clear text, such that $P > B$ and B is primitive root of P . $P = 23, B = 5$	Eve sees $P = 23, B = 5$	Alice and Bob exchange a Prime P and a Base B in clear text, such that $P > B$ and B is primitive root of P . $P = 23, B = 5$
Step 2	Alice generates a random number X_A $X_A = 6$ (keep secret)		Bob generates a random number X_B $X_B = 15$ (keep secret)

Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

Diffie – Hellman: operations

	Alice	Eve	Bob
Step 1	Alice and Bob exchange a Prime P and a Base B in clear text, such that $P > B$ and B is primitive root of P . $P = 23, B = 5$	Eve sees $P = 23, B = 5$	Alice and Bob exchange a Prime P and a Base B in clear text, such that $P > B$ and B is primitive root of P . $P = 23, B = 5$
Step 2	Alice generates a random number X_A $X_A = 6$ (keep secret)		Bob generates a random number X_B $X_B = 15$ (keep secret)
Step 3	Alice computes $Y_A = B^{X_A} \pmod{P}$ $Y_A = 5^6 \pmod{23} = 8$		Bob computes $Y_B = B^{X_B} \pmod{P}$ $Y_B = 5^{15} \pmod{23} = 19$

Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

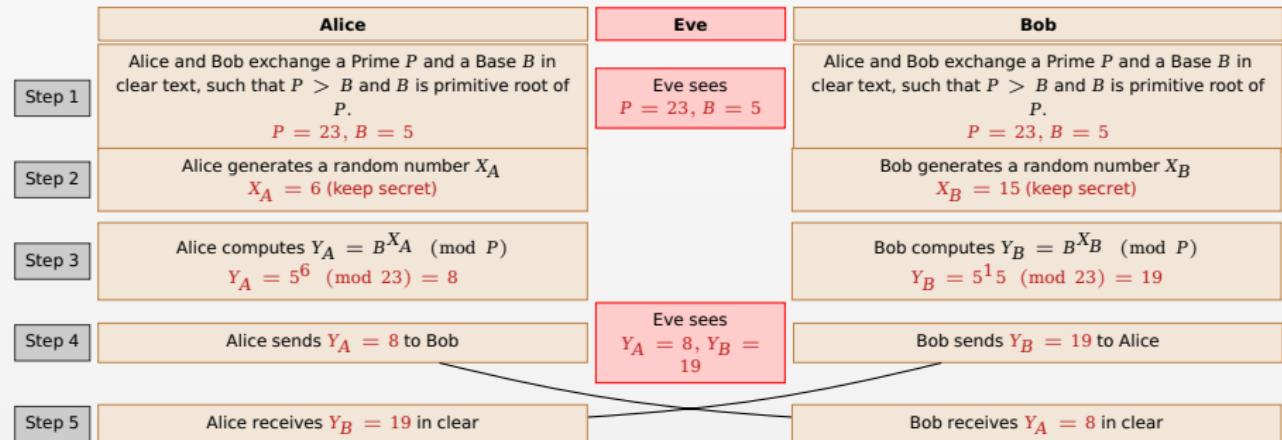
Diffie – Hellman: operations

	Alice	Eve	Bob
Step 1	Alice and Bob exchange a Prime P and a Base B in clear text, such that $P > B$ and B is primitive root of P . $P = 23, B = 5$	Eve sees $P = 23, B = 5$	Alice and Bob exchange a Prime P and a Base B in clear text, such that $P > B$ and B is primitive root of P . $P = 23, B = 5$
Step 2	Alice generates a random number X_A $X_A = 6$ (keep secret)		Bob generates a random number X_B $X_B = 15$ (keep secret)
Step 3	Alice computes $Y_A = B^{X_A} \pmod{P}$ $Y_A = 5^6 \pmod{23} = 8$		Bob computes $Y_B = B^{X_B} \pmod{P}$ $Y_B = 5^{15} \pmod{23} = 19$
Step 4	Alice sends $Y_A = 8$ to Bob	Eve sees $Y_A = 8, Y_B = 19$	Bob sends $Y_B = 19$ to Alice

Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

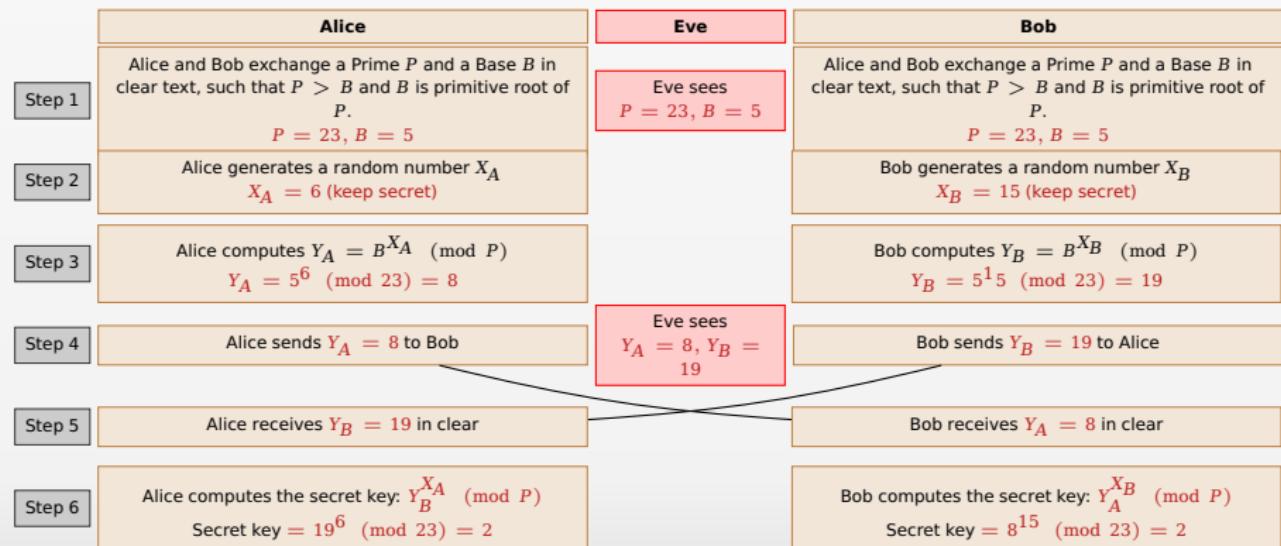
Diffie – Hellman: operations



Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

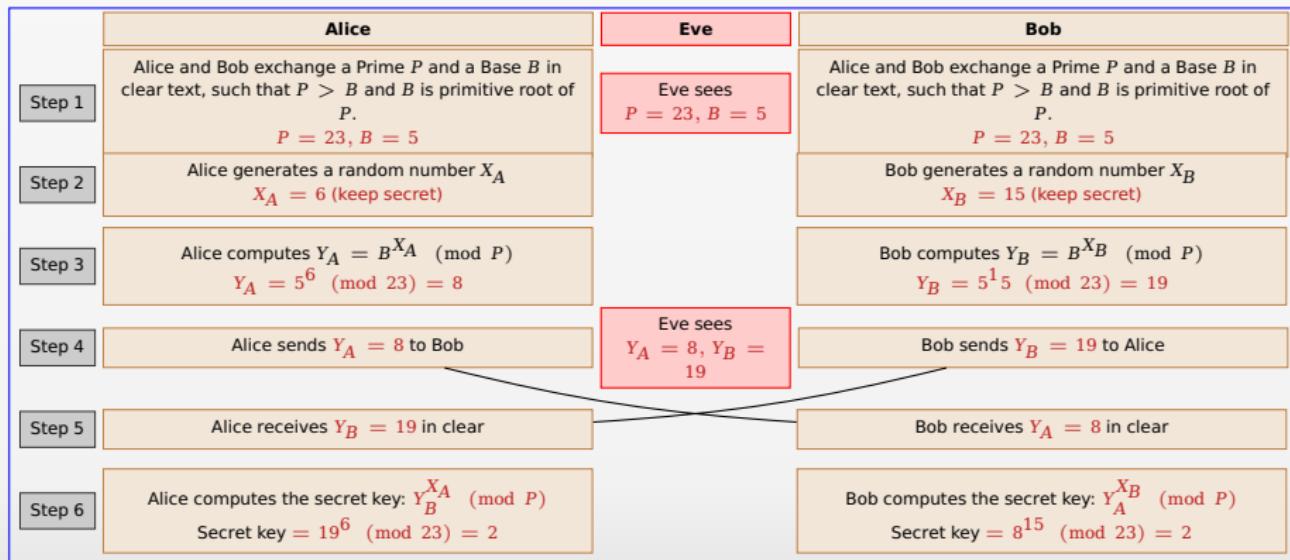
Diffie – Hellman: operations



Modern age: from the end of WW II to our days

Key exchange: Diffie – Hellman

Diffie – Hellman: operations



Modern age: from the end of WW II to our days

Public-key cryptography

Digital signatures

Alice's private key



Alice's public key



Alice

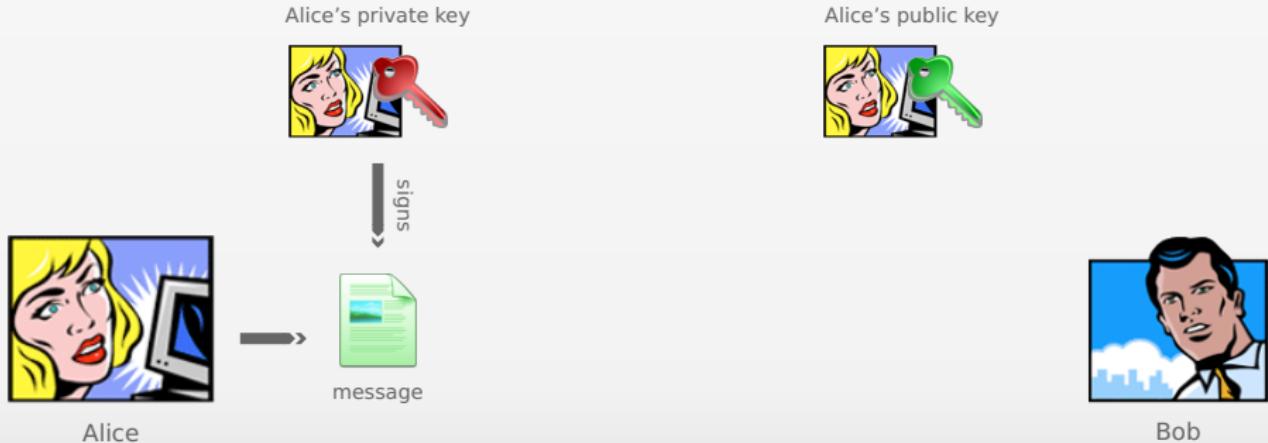


Bob

Modern age: from the end of WW II to our days

Public-key cryptography

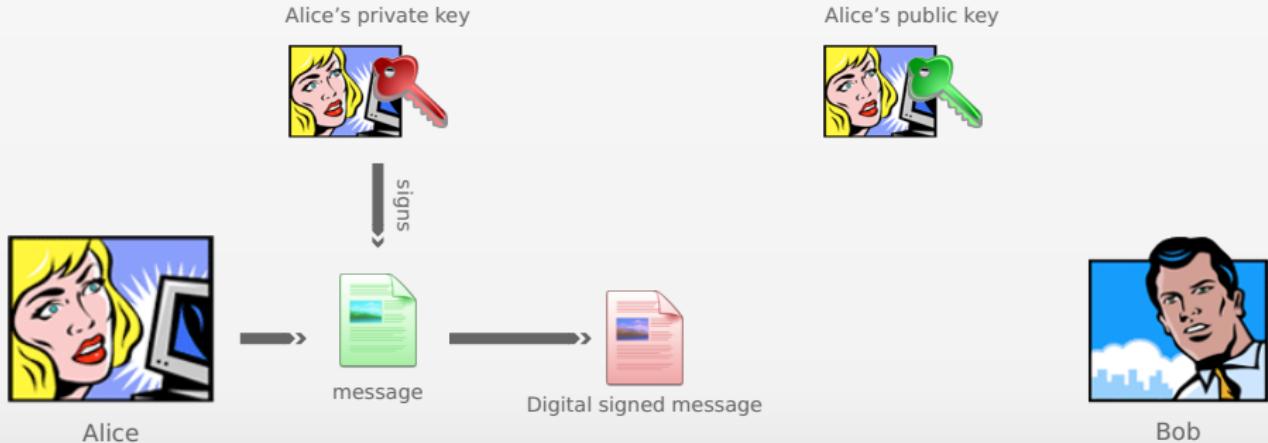
Digital signatures



Modern age: from the end of WW II to our days

Public-key cryptography

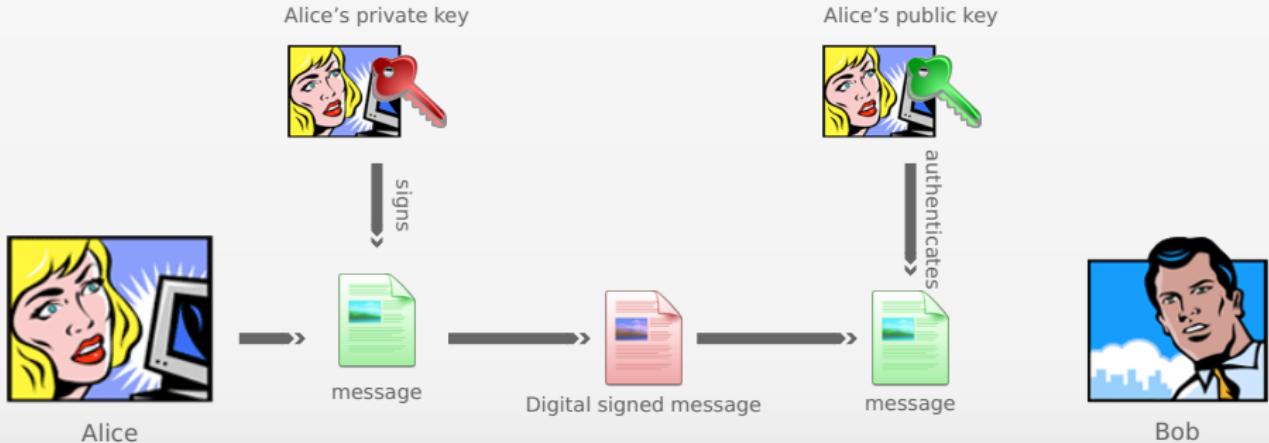
Digital signatures



Modern age: from the end of WW II to our days

Public-key cryptography

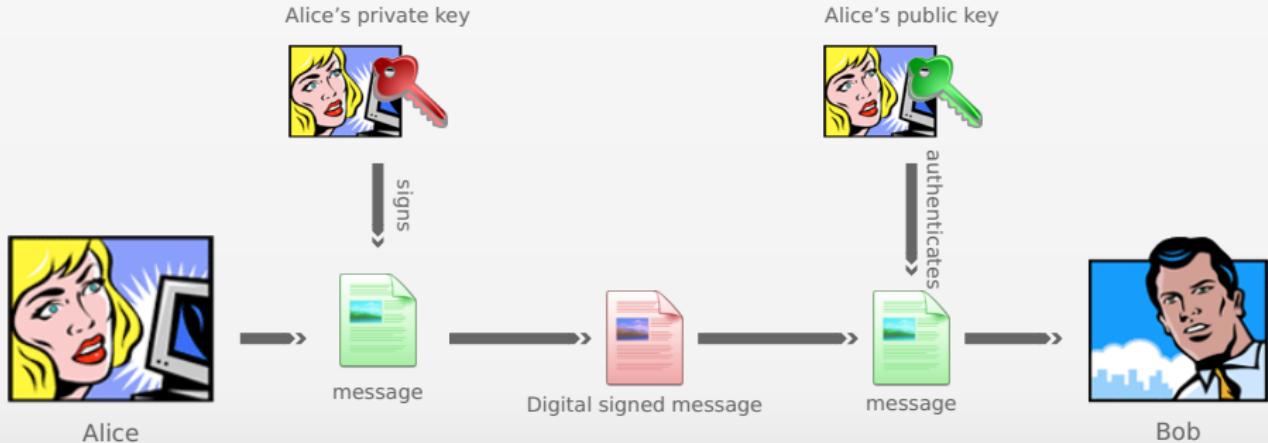
Digital signatures



Modern age: from the end of WW II to our days

Public-key cryptography

Digital signatures



Modern age: from the end of WW II to our days

Public-key cryptography

Encryption and Digital signature

Alice's private key



Bob's public key



Bob's private key



Alice's public key



Alice

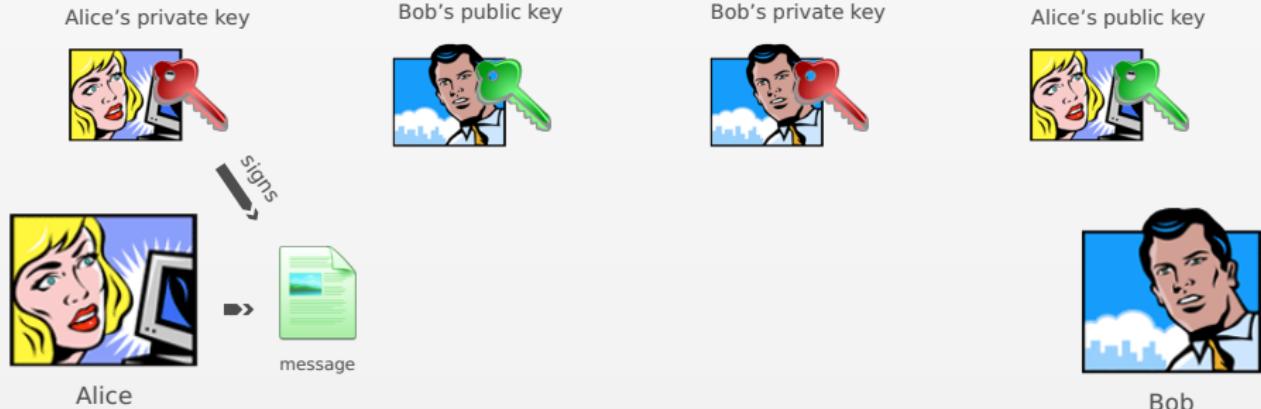


Bob

Modern age: from the end of WW II to our days

Public-key cryptography

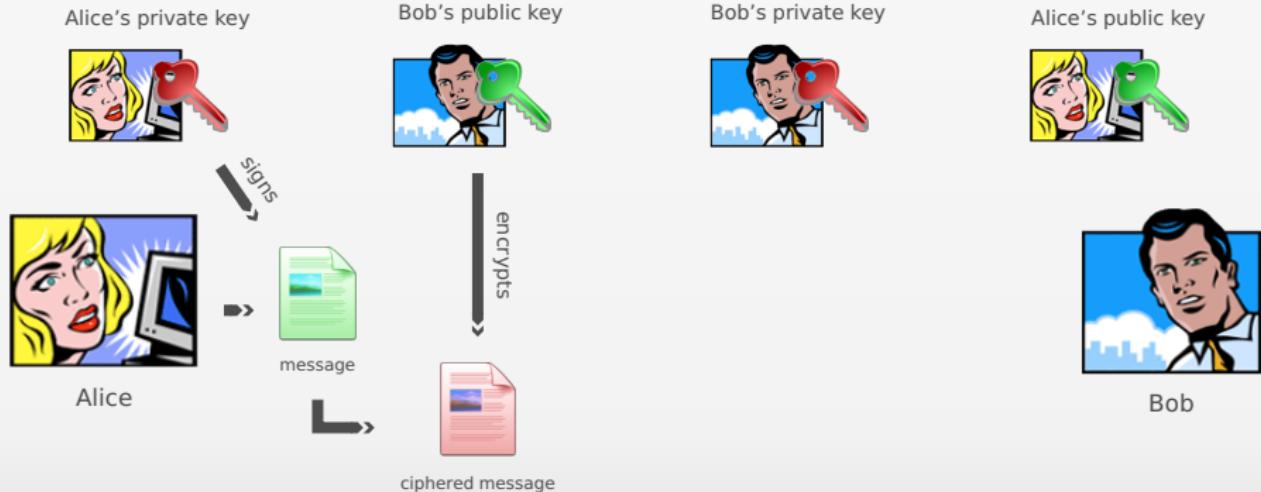
Encryption and Digital signature



Modern age: from the end of WW II to our days

Public-key cryptography

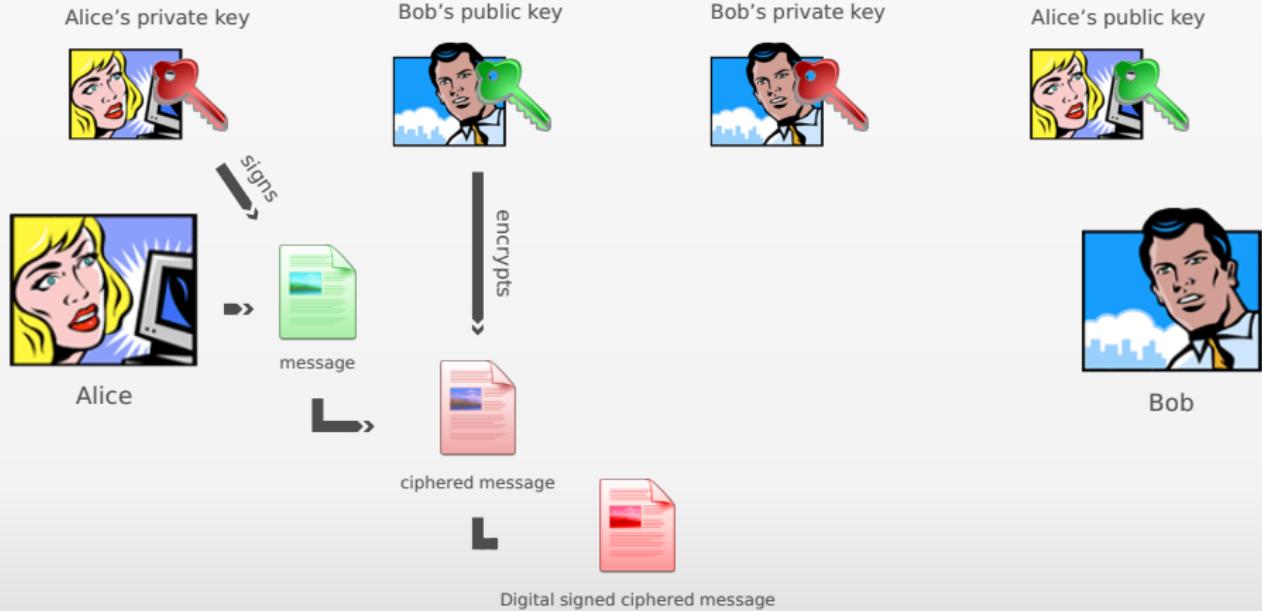
Encryption and Digital signature



Modern age: from the end of WW II to our days

Public-key cryptography

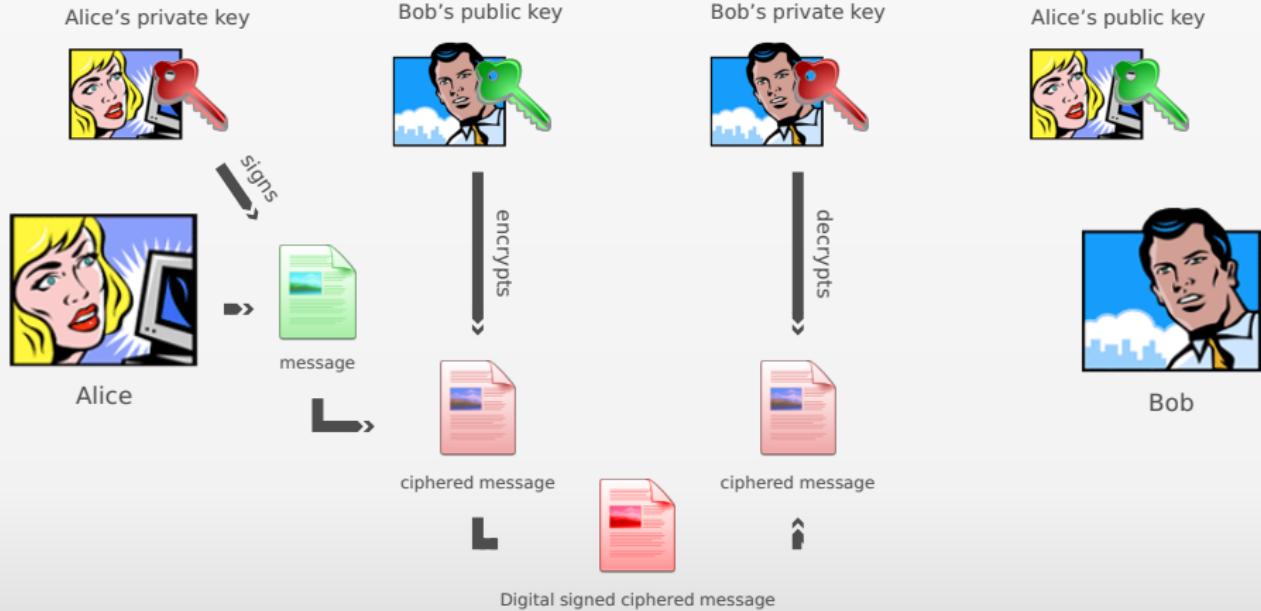
Encryption and Digital signature



Modern age: from the end of WW II to our days

Public-key cryptography

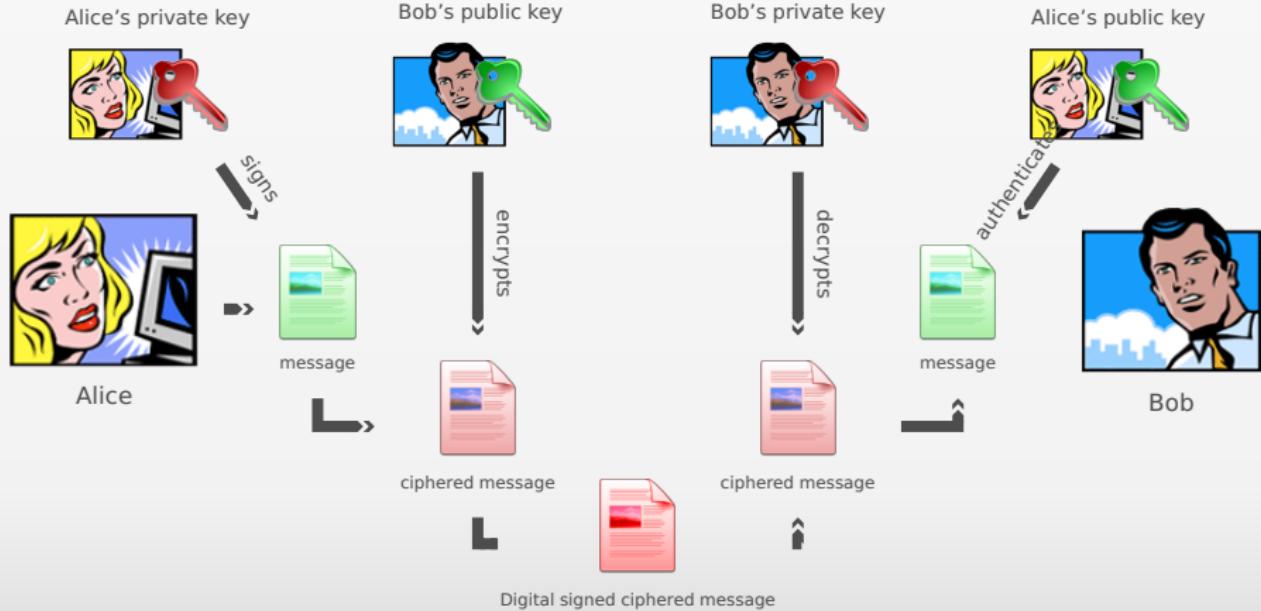
Encryption and Digital signature



Modern age: from the end of WW II to our days

Public-key cryptography

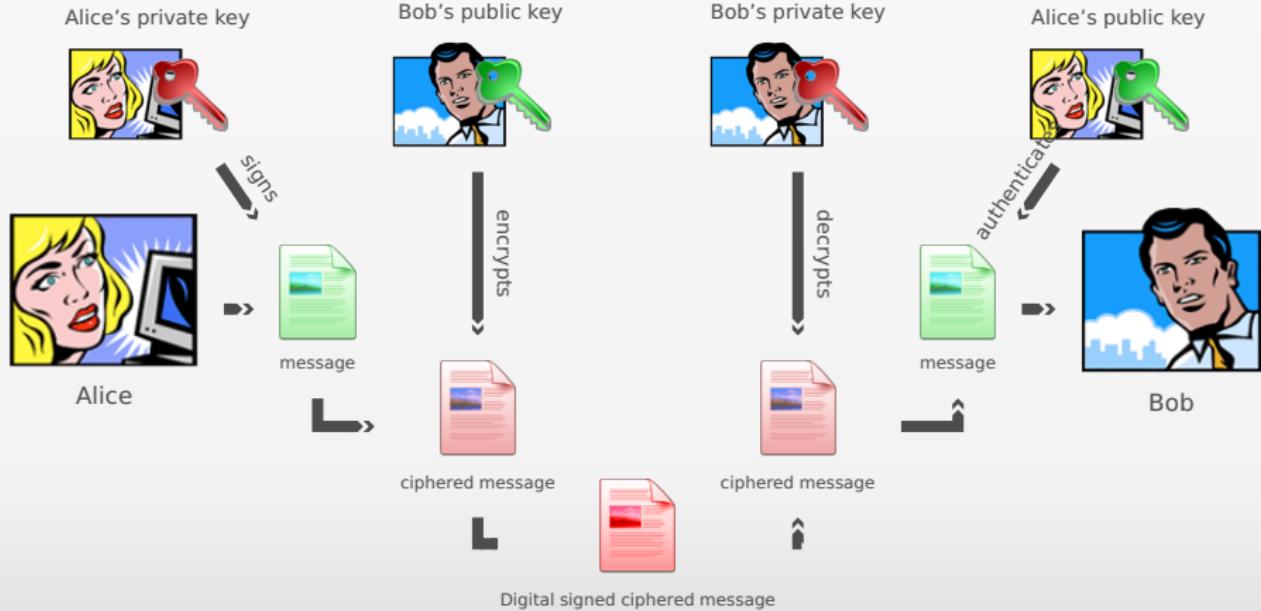
Encryption and Digital signature



Modern age: from the end of WW II to our days

Public-key cryptography

Encryption and Digital signature



Modern age: from the end of WW II to our days

PGP

Pretty Good Privacy



- ▶ **Philipp Zimmermann created the first version of PGP encryption in 1991**
- ▶ Zimmermann was an anti-nuclear activist
- ▶ He created PGP encryption so that people might securely store messages and files
- ▶ No license was required for its non-commercial use and the complete source code was included with all copies
- ▶ PGP has very rapidly acquired a considerable following around the world

Modern age: from the end of WW II to our days

PGP

Pretty Good Privacy



- ▶ **Philipp Zimmermann** created the first version of PGP encryption in 1991
- ▶ **Zimmermann was an anti-nuclear activist**
- ▶ He created PGP encryption so that people might securely store messages and files
- ▶ No license was required for its non-commercial use and the complete source code was included with all copies
- ▶ PGP has very rapidly acquired a considerable following around the world

Modern age: from the end of WW II to our days

PGP

Pretty Good Privacy



- ▶ **Philipp Zimmermann** created the first version of PGP encryption in 1991
- ▶ Zimmermann was an anti-nuclear activist
- ▶ He created PGP encryption so that people might securely store messages and files
- ▶ No license was required for its non-commercial use and the complete source code was included with all copies
- ▶ PGP has very rapidly acquired a considerable following around the world

Modern age: from the end of WW II to our days

PGP

Pretty Good Privacy



- ▶ **Philipp Zimmermann** created the first version of PGP encryption in 1991
- ▶ Zimmermann was an anti-nuclear activist
- ▶ He created PGP encryption so that **people** might securely store messages and files
- ▶ No license was required for its non-commercial use and the complete source code was included with all copies
- ▶ PGP has very rapidly acquired a considerable following around the world

Modern age: from the end of WW II to our days

PGP

Pretty Good Privacy



- ▶ **Philipp Zimmermann** created the first version of PGP encryption in 1991
- ▶ Zimmermann was an anti-nuclear activist
- ▶ He created PGP encryption so that **people might securely store messages and files**
- ▶ No license was required for its non-commercial use and the complete source code was included with all copies
- ▶ PGP has very rapidly acquired a considerable following around the world

Modern age: from the end of WW II to our days

PGP

PGP, OpenPGP and Gnu PG



- ▶ in 1996, Zimmerman created the PGP Inc.
- ▶ In July 1997, PGP Inc. proposed to the IETF that there be a standard called OpenPGP
- ▶ The IETF started the *OpenPGP Working Group*
- ▶ RFC 4880: "*OpenPGP Message Format*"
- ▶ The FSF has developed its own OpenPGP compliant program called **GNU Privacy Guard**
- ▶ In December 1997, PGP Inc. was acquired by Network Associates

Modern age: from the end of WW II to our days

PGP

PGP, OpenPGP and Gnu PG



- ▶ in 1996, Zimmerman created the PGP Inc.
- ▶ In July 1997, PGP Inc. proposed to the IETF that there be a standard called OpenPGP
- ▶ The IETF started the *OpenPGP Working Group*
- ▶ RFC 4880: "*OpenPGP Message Format*"
- ▶ The FSF has developed its own OpenPGP compliant program called *GNU Privacy Guard*
- ▶ In December 1997, PGP Inc. was acquired by Network Associates

Modern age: from the end of WW II to our days

PGP

PGP, OpenPGP and Gnu PG



- ▶ in 1996, Zimmerman created the PGP Inc.
- ▶ In July 1997, PGP Inc. proposed to the IETF that there be a standard called OpenPGP
- ▶ The IETF started the **OpenPGP Working Group**
- ▶ RFC 4880: "*OpenPGP Message Format*"
- ▶ The FSF has developed its own OpenPGP compliant program called **GNU Privacy Guard**
- ▶ In December 1997, PGP Inc. was acquired by **Network Associates**

Modern age: from the end of WW II to our days

PGP

PGP, OpenPGP and Gnu PG



- ▶ in 1996, Zimmerman created the PGP Inc.
- ▶ In July 1997, PGP Inc. proposed to the IETF that there be a standard called OpenPGP
- ▶ The IETF started the **OpenPGP Working Group**
- ▶ RFC 4880: "*OpenPGP Message Format*"
- ▶ The FSF has developed its own OpenPGP compliant program called **GNU Privacy Guard**
- ▶ In December 1997, PGP Inc. was acquired by **Network Associates**
- ▶ In 2010 **Symantec Corp.** acquired PGP Inc.

Modern age: from the end of WW II to our days

PGP

PGP, OpenPGP and Gnu PG



- ▶ in 1996, Zimmerman created the PGP Inc.
- ▶ In July 1997, PGP Inc. proposed to the IETF that there be a standard called OpenPGP
- ▶ The IETF started the **OpenPGP Working Group**
- ▶ RFC 4880: "*OpenPGP Message Format*"
- ▶ **The FSF has developed its own OpenPGP compliant program called GNU Privacy Guard**
- ▶ In December 1997, PGP Inc. was acquired by Network Associates
- ▶ In 2010 Symantec Corp. acquired PGP Inc.

Modern age: from the end of WW II to our days

PGP

PGP, OpenPGP and Gnu PG



- ▶ in 1996, Zimmerman created the PGP Inc.
- ▶ In July 1997, PGP Inc. proposed to the IETF that there be a standard called OpenPGP
- ▶ The IETF started the **OpenPGP Working Group**
- ▶ RFC 4880: "*OpenPGP Message Format*"
- ▶ The FSF has developed its own OpenPGP compliant program called **GNU Privacy Guard**
- ▶ In December 1997, PGP Inc. was acquired by Network Associates
- ▶ In 2010 Symantec Corp. acquired PGP Inc.

Modern age: from the end of WW II to our days

PGP

PGP, OpenPGP and Gnu PG



- ▶ in 1996, Zimmerman created the PGP Inc.
- ▶ In July 1997, PGP Inc. proposed to the IETF that there be a standard called OpenPGP
- ▶ The IETF started the **OpenPGP Working Group**
- ▶ RFC 4880: "*OpenPGP Message Format*"
- ▶ The FSF has developed its own OpenPGP compliant program called **GNU Privacy Guard**
- ▶ In December 1997, PGP Inc. was acquired by **Network Associates**
- ▶ In 2010 **Symantec Corp.** acquired PGP Inc.

Modern age: from the end of WW II to our days

PGP

Practice: Gnu PG

```
# creating a new primary keypair
gpg --gen-key

# generating a revocation certificate
gpg --output revoke.asc --gen-revoke mykeyid

# printing the list of keys
gpg --list-keys

# exporting a key
# in binary format
gpg --output myname.gpg --export mykeyid
# in ASCII format
gpg --output myname.gpg --armor --export mykeyid

# importing a key
gpg --import myfriend.gpg
gpg --list-keys

# validating a key
gpg --edit-key myfriendkeyid
> fpr # print fingerprint of the key
> sign # sign the key
> check # check the key to list the signature
> quit

# encrypt a document
gpg --output invitation.doc.gpg --encrypt \
    --recipient myfriendkeyid invitation.doc

# decrypt a document
gpg --output invitation.doc \
    --decrypt invitation.doc.gpg

# encrypt a document (symmetric version)
gpg --output invitation.doc.gpg --symmetric \
    invitation.doc

# sign a document (detached signature)
gpg --output invitation.doc.sig \
    --detach-sig invitation.doc

# verify the signature
gpg --verify invitation.doc.sig invitation.doc
```

Modern age: from the end of WW II to our days

RSA

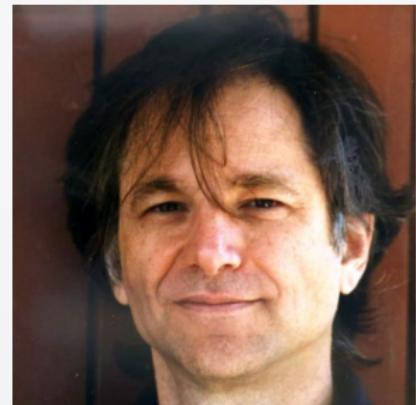
RSA is an algorithm for public-key cryptography



(a) Ron Rivest



(b) Adi Shamir



(c) Leonard Adleman

Figure : The inventors of the RSA algorithm

Modern age: from the end of WW II to our days

RSA - History

- ▶ The RSA algorithm was publicly described in 1978
- ▶ At that time its inventors worked for MIT
- ▶ MIT was granted U.S. Patent 4,405,829 for a "Cryptographic communications system and method" that used the algorithm in 1983

United States Patent [19]		[11] 4,405,829	[45] Sep. 20, 1983
[54]	CRYPTOGRAPHIC COMMUNICATIONS SYSTEM AND METHOD	<i>Primary Examiner</i> —Sal Cangialosi <i>Attorney, Agent, or Firm</i> —Arthur A. Smith, Jr.; Robert J. Horn, Jr.	
[75]	Inventors: Ronald L. Rivest, Belmont; Adi Shamir, Cambridge; Leonard M. Adleman, Arlington, all of Mass.		
[73]	Assignee: Massachusetts Institute of Technology, Cambridge, Mass.		
[21]	Appl. No.: 860,586	[57] ABSTRACT	
[22]	Filed: Dec. 14, 1977		
[51]	Int. Cl. 3	H04K 1/00; H04J 9/04	
[52]	U.S. Cl.	178/22.1; 178/22.11	
[58]	Field of Search	178/22, 22.1, 22.11, 178/22.14, 22.15	
[56]	References Cited		
	U.S. PATENT DOCUMENTS		
	3,657,476 4/1972 Aiken	178/22	

Modern age: from the end of WW II to our days

RSA - History

- ▶ The RSA algorithm was publicly described in **1978**
- ▶ At that time its inventors worked for **MIT**
- ▶ MIT was granted **U.S. Patent 4,405,829** for a "*Cryptographic communications system and method*" that used the algorithm in 1983

United States Patent [19]		[11]	4,405,829
Rivest et al.		[45]	Sep. 20, 1983
[54]	CRYPTOGRAPHIC COMMUNICATIONS SYSTEM AND METHOD		
[75]	Inventors:	Ronald L. Rivest, Belmont; Adi Shamir, Cambridge; Leonard M. Adleman, Arlington, all of Mass.	Primary Examiner—Sal Cangialosi Attorney, Agent, or Firm—Arthur A. Smith, Jr.; Robert J. Horn, Jr.
[73]	Assignee:	Massachusetts Institute of Technology, Cambridge, Mass.	
[21]	Appl. No.:	860,586	[57] ABSTRACT
[22]	Filed:	Dec. 14, 1977	A cryptographic communications system and method. The system includes a communications channel coupled to at least one terminal having an encoding device and to at least one terminal having a decoding device. A message-to-be-transferred is enciphered to ciphertext at the encoding terminal by first encoding the message as a number M in a predetermined set, and then raising that number to a first predetermined power (associated with the intended receiver) and finally computing the remainder, or residue, C, when the exponentiated number is divided by the product of two predetermined prime numbers (associated with the intended receiver). The residue C is the ciphertext. The ciphertext is deciphered to the original message at the decoding terminal in a
[51]	Int. Cl.:	H04K 1/00; H04J 9/04	
[52]	U.S. Cl.:	178/22.1; 178/22.11	
[58]	Field of Search:	178/22, 22.1, 22.11, 178/22.14, 22.15	
[56]	References Cited		
	U.S. PATENT DOCUMENTS		
3,657,476	4/1972 Aiken	178/22

Modern age: from the end of WW II to our days

RSA - History

- ▶ The RSA algorithm was publicly described in **1978**
- ▶ At that time its inventors worked for **MIT**
- ▶ MIT was granted **U.S. Patent 4,405,829** for a "*Cryptographic communications system and method*" that used the algorithm in 1983

United States Patent [19]		[11] 4,405,829
Rivest et al.		[45] Sep. 20, 1983
[54]	CRYPTOGRAPHIC COMMUNICATIONS SYSTEM AND METHOD	
[75]	Inventors:	Ronald L. Rivest, Belmont; Adi Shamir, Cambridge; Leonard M. Adleman, Arlington, all of Mass.
[73]	Assignee:	Massachusetts Institute of Technology, Cambridge, Mass.
[21]	Appl. No.:	860,586
[22]	Filed:	Dec. 14, 1977
[51]	Int. Cl. ³	H04K 1/00; H04J 9/04
[52]	U.S. Cl.	178/22.1; 178/22.11
[58]	Field of Search	178/22, 22.1, 22.11, 178/22.14, 22.15
[56]	References Cited	
	U.S. PATENT DOCUMENTS	
3,657,476	4/1972 Aiken	178/22

*Primary Examiner—Sal Cangialosi
Attorney, Agent, or Firm—Arthur A. Smith, Jr.; Robert J. Horn, Jr.*

[57] ABSTRACT

A cryptographic communications system and method. The system includes a communications channel coupled to at least one terminal having an encoding device and to at least one terminal having a decoding device. A message-to-be-transferred is enciphered to ciphertext at the encoding terminal by first encoding the message as a number M in a predetermined set, and then raising that number to a first predetermined power (associated with the intended receiver) and finally computing the remainder, or residue, C, when the exponentiated number is divided by the product of two predetermined prime numbers (associated with the intended receiver). The residue C is the ciphertext. The ciphertext is deciphered to the original message at the decoding terminal in a

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $m^e \pmod{n} \rightarrow 65^{17} \pmod{3233} = 2790$
- ▶ Bob sends the ciphertext $c = 2790$ to Alice

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $m^e \pmod{n} \rightarrow 65^{17} \pmod{3233} \approx 65$
- ▶ Bob sends the message 65 to Alice

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 65^{2753} \pmod{3233} \approx 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $m^e \pmod{n} \rightarrow 65^{17} \pmod{3233} \approx 65$
- ▶ Bob sends the message 65

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 65^{2753} \pmod{3233} \approx 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $m^e \pmod{n} \rightarrow 65^{17} \pmod{3233} \approx 65$
- ▶ Bob sends the message 65

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 65^{2753} \pmod{3233} \approx 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

Bob wants to send the message m to Alice $\rightarrow m = 65$

The public key of Alice is $(n, e) \rightarrow (3233, 17)$

Bob computes $m^e \pmod{n} \rightarrow 65^{17} \pmod{3233} \approx 2753$

This value is the cipher text

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2753^{2753} \pmod{3233} \approx 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

Alice wants to send the message $m = 65 \rightarrow m = 65$

The public key of Bob is $(n, e) \rightarrow (3233, 17)$

Bob computes $m^e \pmod{n} \rightarrow m^e = 65^{17} \pmod{3233} \approx 2753$

Bob sends the ciphertext $c = 2753$

► Decryption

- ▶ Alice computes $c^d \pmod{n} \rightarrow c^d = 2753^{2753} \pmod{3233} \approx 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$

The public key of Alice is $(n, e) \rightarrow (3233, 17)$

Bob computes $m^e \pmod{n} \rightarrow 65^{17} \pmod{3233} \approx 2790$

This value is sent to Alice via a public channel.

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} \approx 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$

Bob computes $m^e \pmod{n} \rightarrow 65^{17} \pmod{3233} \approx 2790$

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow 2790^{2753} \pmod{3233} \approx 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice receives $c \rightarrow c = 2790$
- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice receives $c \rightarrow c = 2790$
- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice decodes the message $\rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice receives $c \rightarrow c = 2790$
- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice receives $c \rightarrow c = 2790$
- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

RSA - Operation

► Key generation

- ▶ Choose two distinct prime numbers p and $q \rightarrow p = 61, q = 53$
- ▶ Compute $n = pq \rightarrow n = 3233$
- ▶ Compute $\phi(n) = (p - 1)(q - 1)$ (ϕ is the Euler totient) $\rightarrow \phi(3233) = 3120$
- ▶ Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1 \rightarrow e = 17$
- ▶ the public key is $(n, e) \rightarrow (3233, 17)$
- ▶ Compute $de = 1 \pmod{\phi(n)} \rightarrow d = 2753$
- ▶ the private key is $(n, d) \rightarrow (3233, 2753)$

► Encryption

- ▶ Bob wants to send the message m to Alice $\rightarrow m = 65$
- ▶ The public key of Alice is $(n, e) \rightarrow (3233, 17)$
- ▶ Bob computes $c = m^e \pmod{n} \rightarrow c = 65^{17} \pmod{3233} = 2790$
- ▶ Bob transmits c to Alice $\rightarrow c = 2790$

► Decryption

- ▶ Alice receives $c \rightarrow c = 2790$
- ▶ Alice computes $m = c^d \pmod{n} \rightarrow m = 2790^{2753} \pmod{3233} = 65$
- ▶ Alice reads the message $m \rightarrow m = 65$

Modern age: from the end of WW II to our days

Tutorial: RSA

- ▶ Using Sage, write a program which implement RSA:
 - ▶ generation of private and public keys (*for calculating d such that de = 1 (mod φ(n)) you may use the extended Euclidean algorithm → (xgcd) command in sage*)
 - ▶ p and q are fixed
 - ▶ $p = (2^{31}) - 1$
 - ▶ $q = (2^{61}) - 1$
 - ▶ encryption of a message giving the public key
 - ▶ decryption of a cipher giving the private key

Modern age: from the end of WW II to our days

RSA

Tutorial: RSA → a solution

```
# generating p, q, n
p = (2^31) - 1; p
is_prime(p)
q = (2^61) - 1; q
is_prime(q)
n = p * q ; n

# generating e
e = ZZ.random_element(euler_phi(n))
while gcd(e, euler_phi(n)) != 1:
    e = ZZ.random_element(euler_phi(n))
e
e<n

# generating d
bezout = xgcd(e, euler_phi(n)) ; bezout
d = Integer(mod(bezout[1], euler_phi(n)))
mod(d * e, euler_phi(n))

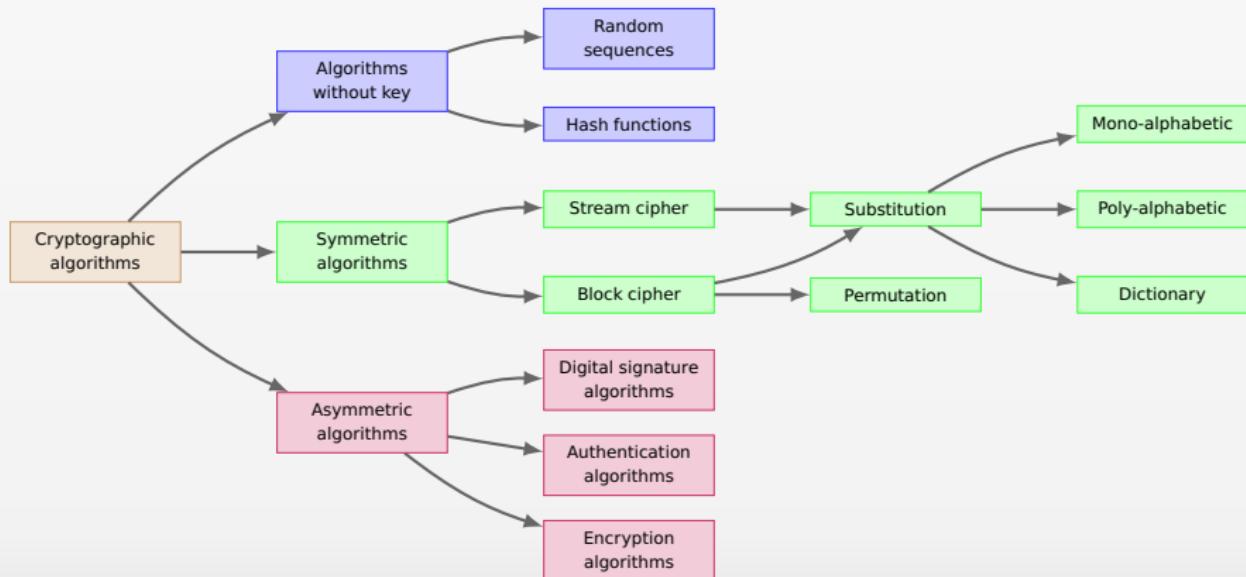
# define modular exponentiation function
def modexp(a, b, n):
    d=1
    for i in list(Integer.binary(b)):
        d = mod(d * d, n)
    if Integer(i) == 1:
        d = mod(d * a, n)
    return Integer(d)

# encryption
m = "HELLOWORLD"
m = [ord(c) for c in m]; m
m = ZZ(list(reversed(m)), 100); m
c = modexp(m, e, n); c

# decryption
dc = modexp(c, d, n); dc
m == dc
```

Modern age: from the end of WW II to our days

Taxonomy of cryptographic algorithms



In practice...

- 1** Introduction
- 2** Mathematical background
- 3** History
- 4** In practice...



The Web

- 4** In practice...
- The Web
 - The Emails
 - Hash functions
 - Data



Stream cipher

Connection in clear on www.google.fr

```
GET / HTTP/1.1
Host: www.google.fr
User-Agent: Mozilla/5.0 (X11; U; Linux i686; fr; rv:1.8.1.11) Gecko/20071127
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/p
Accept-Language: fr,en-us;q=0.7,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-15,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Set-Cookie: PREF=ID=0fb800987e6aebe1:TM=1206392434:LM=1206392434:S=00BbJFYHLC
Content-Encoding: gzip
Server: gws
Content-Length: 2733
Date: Mon, 24 Mar 2008 21:00:34 GMT
```

Stream cipher

Ciphered connection through SSL on www.google.fr

```
.....:=....  
...(....F#..!.....r].....8.  
...9.8.....5.....3.2...../.....  
.....  
...(...  
www.google.fr.  
.....P...L...G.....v...oQp.D.M.Lj.? ..m...R... .`...?...Y.&f.....  
..*.H..  
....OL1.0...U....ZA1%0#..U.  
.Thawte Consulting (Pty) Ltd.1.0...U...  
Thawte SGC CAO..  
070503153458Z.  
08051423181120h1.0...U....US1.0...U...  
Californial.0...U...  
Mountain View1.0...U.  
.  
Google Incl.0...U....www.google.com0..0  
..*.H..  
.....0.....F....s4H.ud...v'r....;.....V...sh....$.0...6].sc  
.....,T.....0..0(..U.%!0...+.....+.....`..H...B..06..U.../0-+.  
..*.H..  
.....}...2.;!..U.....+Tz%..s.|  
..z.\.#PW.l.[...ys...vk....V.....vKd....+ ....2.|.bw....!....1....Az....V....c  
+ ..
```

Stream cipher

Protocols proposed by www.google.fr

```
SSL handshake has read 1777 bytes and written 316 bytes
---
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 1024 bit
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol : TLSv1
    Cipher   : AES256-SHA
    Session-ID: B40A7F0C631658D70DCFFF3F94B513598E430FAD56C77599A
    Session-ID-ctx:
    Master-Key: 83BABE8C038466DEC7301EFCC8D17B34D665F025950A0F228
    Key-Aggr  : None
    Start Time: 1206392348
    Timeout   : 300 (sec)
    Verify return code: 20 (unable to get local issuer certificate)
---
read:errno=0
```

The Emails

4 In practice...

- The Web
- The Emails**
- Hash functions
- Data



Encryption of emails

Example of encrypted email

—BEGIN PGP MESSAGE—

Charset: ISO-8859-1

Version: GnuPG v1.4.6 (GNU/Linux)

hQEAOA+6Lqc04sm0qEAP/d0yeNbAxiyPdsyuwe8zI8U4RH9TOzecs7m3+Nu5MXLd
QxH5yyIw5opwLFEn+ZrH8E2xvf5V6G44z/WzSE+X8EYU8b3fWIBqTmALiTZI+/Q
B5Qn60NwXoyWeCdio75bt4KChYdxEuJYfQIsO9gyjg0jvvUpAKpiC5hMUFUne7YD
/0lppa2lqgcbqbGwfQsZgaRrt10hlcFkGw0pa3lqCT8ydoynkDptK5lhCoHxo0
+GJNhVCAMA7/+oXnLWZIje+p/JjZSunk4YD0sQnR08hAUDf106ul3nKj8/D3Ojgp
ZaoUHe/1y7t0LcBDfhXcrZY3agNZbeDzTYgHWEeUa6qjXkZltdwRfbmtoqLYlj9
zPeS/ecFutXjlJmlJvwuqecnWx3iLTVk15ApTfe+djKPW9iPsg4XWoO+8gmcTru3M
fuk7WcRsluNR8jW83H+oNZVxVmRAOmNXyeDworVRYfvjOGCipEkgsHLAQ/6OURG
98g7Av8uvRsN39k9wgi2Q4isIBGX5oK1e/XpS4EuXnfQ7NyheXhiWBMrkdgZAp6
h4jnd4OWxGjjnujqTTKwKUfoqaThgxosio+MLAGb7BqUrqMBgr3OaUccNwZvoA86
SAG3Ve0Ndysl2htwMEs4y2cNplrcXME2/rPR4vPiOq0q3pYwBeATTrfISPwgw
XKCiuInWgZjZ2ggW1ip0219HGzOxg8mKVWEIYrPRwqKKRk6X5xwldo==yyg5

—END PGP MESSAGE—

Hash functions

- 4** In practice...
- The Web
 - The Emails
 - Hash functions**
 - Data



Hash functions

Definition

A hash function is a **one-way mathematical** function which, from an original text, computes a **unique fingerprint** of fixed size and irreversible.

- ▶ **TOTO** → 04c1d7cd203ef496f200ee5a096b5764
- ▶ **ToTo** → 3cca12013a4f82de305ba73b01a84509
- ▶ **Toto** → 998db284485ec6c227f8dc34086128e1
- ▶ **toto** → f71dbe52628a3f83a77ab494817525c6
- ▶ **The sky is red it will be fine tomorrow.** →
752347b3a885922ba1be45ecb665917d

Hash functions are especially used to store passwords

Hash functions

Definition

A hash function is a **one-way mathematical** function which, from an original text, computes a **unique fingerprint** of fixed size and irreversible.

- ▶ **TOTO** \Longrightarrow 04c1d7cd203ef496f200ee5a096b5764
- ▶ **ToTo** \Longrightarrow 3cca12013a4f82de305ba73b01a84509
- ▶ **Toto** \Longrightarrow 998db284485ec6c227f8dc34086128e1
- ▶ **toto** \Longrightarrow f71dbe52628a3f83a77ab494817525c6
- ▶ **The sky is red It will be fine tomorrow.** \Longrightarrow
752347b3a885922ba1be45ecb665917d

Hash functions are especially used to store passwords

Hash functions

Definition

A hash function is a **one-way mathematical** function which, from an original text, computes a **unique fingerprint** of fixed size and irreversible.

- ▶ **TOTO** \Longrightarrow 04c1d7cd203ef496f200ee5a096b5764
- ▶ **ToTo** \Longrightarrow 3cca12013a4f82de305ba73b01a84509
- ▶ **Toto** \Longrightarrow 998db284485ec6c227f8dc34086128e1
- ▶ **toto** \Longrightarrow f71dbe52628a3f83a77ab494817525c6
- ▶ **The sky is red it will be fine tomorrow.** \Longrightarrow
752347b3a885922ba1be45ecb665917d

Hash functions are especially used to store passwords

Hash functions

Definition

A hash function is a **one-way mathematical** function which, from an original text, computes a **unique fingerprint** of fixed size and irreversible.

- ▶ **TOTO** \Longrightarrow 04c1d7cd203ef496f200ee5a096b5764
- ▶ **ToTo** \Longrightarrow 3cca12013a4f82de305ba73b01a84509
- ▶ **Toto** \Longrightarrow 998db284485ec6c227f8dc34086128e1
- ▶ **toto** \Longrightarrow f71dbe52628a3f83a77ab494817525c6
- ▶ **The sky is red it will be fine tomorrow.** \Longrightarrow
752347b3a885922ba1be45ecb665917d

Hash functions are especially used to store passwords

Hash functions

Definition

A hash function is a **one-way mathematical** function which, from an original text, computes a **unique fingerprint** of fixed size and irreversible.

- ▶ **TOTO** \Longrightarrow 04c1d7cd203ef496f200ee5a096b5764
- ▶ **ToTo** \Longrightarrow 3cca12013a4f82de305ba73b01a84509
- ▶ **Toto** \Longrightarrow 998db284485ec6c227f8dc34086128e1
- ▶ **toto** \Longrightarrow f71dbe52628a3f83a77ab494817525c6
- ▶ **The sky is red it will be fine tomorrow.** \Longrightarrow
752347b3a885922ba1be45ecb665917d

Hash functions are especially used to store passwords

Hash functions

Definition

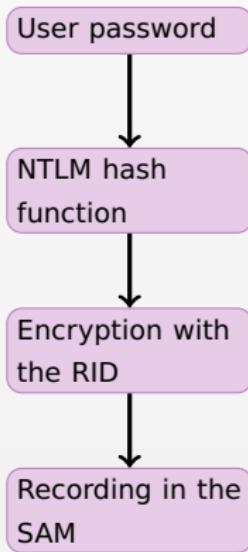
A hash function is a **one-way mathematical** function which, from an original text, computes a **unique fingerprint** of fixed size and irreversible.

- ▶ **TOTO** \Longrightarrow 04c1d7cd203ef496f200ee5a096b5764
- ▶ **ToTo** \Longrightarrow 3cca12013a4f82de305ba73b01a84509
- ▶ **Toto** \Longrightarrow 998db284485ec6c227f8dc34086128e1
- ▶ **toto** \Longrightarrow f71dbe52628a3f83a77ab494817525c6
- ▶ **The sky is red it will be fine tomorrow.** \Longrightarrow
752347b3a885922ba1be45ecb665917d

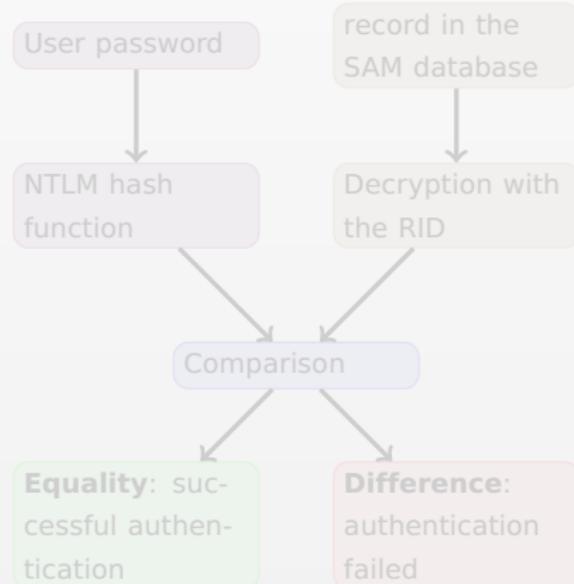
Hash functions are especially used to store passwords

Passwords

Authentication – Local authentication process on Windows



Storage process

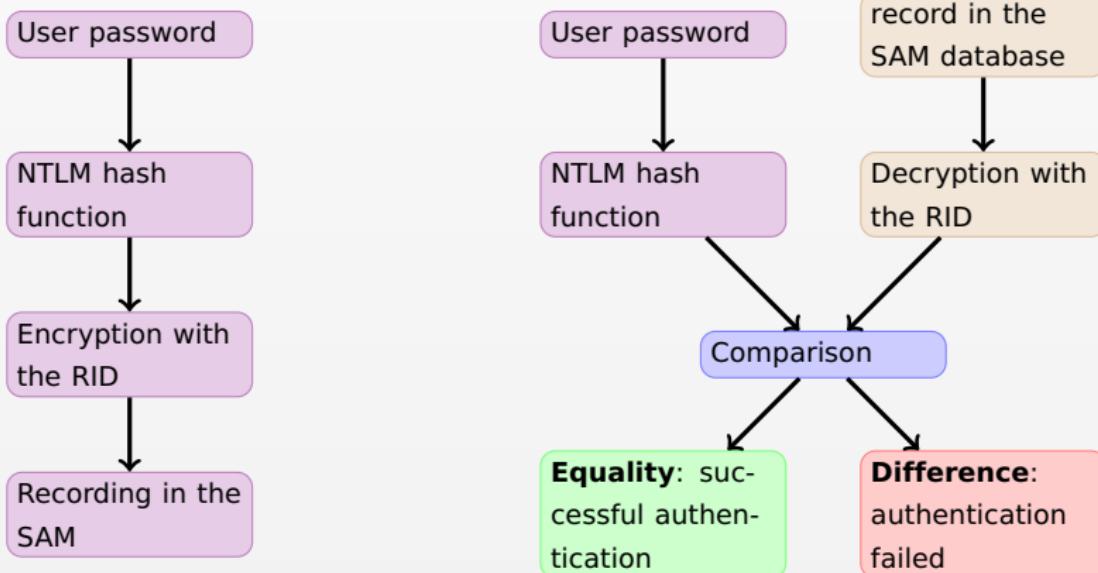


Authentication process

Source: <http://www.hsc.fr/>

Passwords

Authentication – Local authentication process on Windows



Storage process

Authentication process

Source: <http://www.hsc.fr/>

Passwords

Find a password

Guess the password

With **social engineering** techniques (*hoax calls, scavenging, Internet search...*).

Spoof the system

With **spoofing** techniques *Fake authentication software.*

Spy the system

With **keylogging** techniques or by network **sniffing**.

Attack the password file

With techniques of attacks by **dictionary** or by **brute force**.

Passwords

Find a password

Guess the password

With **social engineering** techniques (*hoax calls, scavenging, Internet search...*).

Spoof the system

With **spoofing** techniques *Fake authentication software.*

Spy the system

With **keylogging** techniques or by network **sniffing**.

Attack the password file

With techniques of attacks by **dictionary** or by **brute force**.

Passwords

Find a password

Guess the password

With **social engineering** techniques (*hoax calls, scavenging, Internet search...*).

Spoof the system

With **spoofing** techniques *Fake authentication software.*

Spy the system

With **keylogging** techniques or by network **sniffing**.

Attack the password file

With techniques of attacks by **dictionary** or by **brute force**.

Passwords

Find a password

Guess the password

With **social engineering** techniques (*hoax calls, scavenging, Internet search...*).

Spoof the system

With **spoofing** techniques *Fake authentication software.*

Spy the system

With **keylogging** techniques or by network **sniffing**.

Attack the password file

With techniques of attacks by **dictionary** or by **brute force**.

Passwords

Find a password

Attack by dictionary

- for each word in a dictionary
- computing of the hash
- comparison with that recorded in the system



151349	michaella
151350	michaelmas
151351	michaelmastide
151352	michail
151353	michal
151354	michale
151355	miche
151356	micheal
151357	micheil
151358	michel
151359	michelangelesque
151360	michelangelism
151361	michelangelo
151362	michele
151363	michelia
151364	michelin
151365	michelina
151366	micheline
151367	michell
151368	michelle
151369	michelson
151370	micher

Passwords

Find a password

Attack by dictionary

- ➊ for each word in a dictionary
- ➋ computing of the hash
- ➌ comparison with that recorded in the system



151349	michaella
151350	michaelmas
151351	michaelmastide
151352	michail
151353	michal
151354	michale
151355	miche
151356	micheal
151357	micheil
151358	michel
151359	michelangelesque
151360	michelangelism
151361	michelangelo
151362	michele
151363	michelia
151364	michelin
151365	michelina
151366	micheline
151367	michell
151368	michelle
151369	michelson
151370	micher

Passwords

Find a password

Attack by dictionary

- ➊ for each word in a dictionary
- ➋ computing of the hash
- ➌ comparison with that recorded in the system



151349	michaella
151350	michaelmas
151351	michaelmastide
151352	michail
151353	michal
151354	michale
151355	miche
151356	micheal
151357	micheil
151358	michel
151359	michelangelesque
151360	michelangelism
151361	michelangelo
151362	michele
151363	michelia
151364	michelin
151365	michelina
151366	micheline
151367	michell
151368	michelle
151369	michelson
151370	micher

Passwords

Find a password

Attack by dictionary

- ➊ for each word in a dictionary
- ➋ computing of the hash
- ➌ comparison with that recorded in the system



151349	michaella
151350	michaelmas
151351	michaelmastide
151352	michail
151353	michal
151354	michale
151355	miche
151356	micheal
151357	micheil
151358	michel
151359	michelangelesque
151360	michelangelism
151361	michelangelo
151362	michele
151363	michelia
151364	michelin
151365	michelina
151366	micheline
151367	michell
151368	michelle
151369	michelson
151370	micher

Passwords

Find a password

Attack by brute force

- ▶ browse the exhaustive space of passwords
- ▶ more the password is long, more it is difficult to find it
- ▶ more the passwords space is long, more time to browse it will be long

	Lowercase letters (26)	Alphanumeric characters (62)	ASCII characters (256)
4 characters	$26^4 = 460000$	$62^4 = 1,5 \cdot 10^7$	$256^4 = 4,3 \cdot 10^9$
5 characters	$26^5 = 1,2 \cdot 10^7$	$62^5 = 9,2 \cdot 10^8$	$256^5 = 1,1 \cdot 10^{12}$
6 characters	$26^6 = 3,1 \cdot 10^8$	$62^6 = 5,7 \cdot 10^{10}$	$256^6 = 2,8 \cdot 10^{14}$
7 characters	$26^7 = 8,0 \cdot 10^9$	$62^7 = 3,5 \cdot 10^{12}$	$256^7 = 7,2 \cdot 10^{16}$
8 characters	$26^8 = 2,1 \cdot 10^{11}$	$62^8 = 2,2 \cdot 10^{14}$	$256^8 = 1,8 \cdot 10^{19}$

Passwords

Find a password

Attack by brute force

- ▶ browse the exhaustive space of passwords
- ▶ more the password is long, more it is difficult to find it
- ▶ more the passwords space is long, more time to browse it will be long

	Lowercase letters (26)	Alphanumeric characters (62)	ASCII characters (256)
4 characters	$26^4 = 460000$	$62^4 = 1,5 \cdot 10^7$	$256^4 = 4,3 \cdot 10^9$
5 characters	$26^5 = 1,2 \cdot 10^7$	$62^5 = 9,2 \cdot 10^8$	$256^5 = 1,1 \cdot 10^{12}$
6 characters	$26^6 = 3,1 \cdot 10^8$	$62^6 = 5,7 \cdot 10^{10}$	$256^6 = 2,8 \cdot 10^{14}$
7 characters	$26^7 = 8,0 \cdot 10^9$	$62^7 = 3,5 \cdot 10^{12}$	$256^7 = 7,2 \cdot 10^{16}$
8 characters	$26^8 = 2,1 \cdot 10^{11}$	$62^8 = 2,2 \cdot 10^{14}$	$256^8 = 1,8 \cdot 10^{19}$

Passwords

Find a password

Attack by brute force

- ▶ browse the exhaustive space of passwords
- ▶ more the password is long, more it is difficult to find it
- ▶ more the passwords space is long, more time to browse it will be long

	Lowercase letters (26)	Alphanumeric characters (62)	ASCII characters (256)
4 characters	$26^4 = 460000$	$62^4 = 1,5 \cdot 10^7$	$256^4 = 4,3 \cdot 10^9$
5 characters	$26^5 = 1,2 \cdot 10^7$	$62^5 = 9,2 \cdot 10^8$	$256^5 = 1,1 \cdot 10^{12}$
6 characters	$26^6 = 3,1 \cdot 10^8$	$62^6 = 5,7 \cdot 10^{10}$	$256^6 = 2,8 \cdot 10^{14}$
7 characters	$26^7 = 8,0 \cdot 10^9$	$62^7 = 3,5 \cdot 10^{12}$	$256^7 = 7,2 \cdot 10^{16}$
8 characters	$26^8 = 2,1 \cdot 10^{11}$	$62^8 = 2,2 \cdot 10^{14}$	$256^8 = 1,8 \cdot 10^{19}$

Passwords

Find a password

Attack by brute force

- ▶ browse the exhaustive space of passwords
- ▶ more the password is long, more it is difficult to find it
- ▶ more the passwords space is long, more time to browse it will be long

	Lowercase letters (26)	Alphanumeric characters (62)	ASCII characters (256)
4 characters	$26^4 = 460000$	$62^4 = 1,5 \cdot 10^7$	$256^4 = 4,3 \cdot 10^9$
5 characters	$26^5 = 1,2 \cdot 10^7$	$62^5 = 9,2 \cdot 10^8$	$256^5 = 1,1 \cdot 10^{12}$
6 characters	$26^6 = 3,1 \cdot 10^8$	$62^6 = 5,7 \cdot 10^{10}$	$256^6 = 2,8 \cdot 10^{14}$
7 characters	$26^7 = 8,0 \cdot 10^9$	$62^7 = 3,5 \cdot 10^{12}$	$256^7 = 7,2 \cdot 10^{16}$
8 characters	$26^8 = 2,1 \cdot 10^{11}$	$62^8 = 2,2 \cdot 10^{14}$	$256^8 = 1,8 \cdot 10^{19}$

Data

4 In practice...

- The Web
- The Emails
- Hash functions
- Data**



File encryption

File encryption with AES

Encryption

```
openssl enc -e -aes-256-cbc -k password -in hamlet.txt -out hamlet-cipher.txt
```

Decryption

```
openssl enc -d -aes-256-cbc -k password -in hamlet-cipher.txt
```



hamlet.txt - Data

```
Long: $000005D0 | Type/Créateur: / | Sér: $00000000:$00000000:$00000000
00000000: 00 6F 28 62 65 2C 2B 6F 72 2B 6E 0F 74 20 74 66 To be, or not to
00000010: 28 62 65 3B 2B 74 68 61 74 2B 69 73 2B 74 68 65 be; that is the
00000018: 73 21 75 65 73 74 69 66 6E 3B 0R 57 68 65 74 66 question: Whether
00000028: 65 72 2B 27 74 69 73 2B 6E 6F 62 6C 65 72 2B 69 er 'tis nobler i
00000038: 66 2B 74 68 65 2B 6D 69 66 6E 2B 74 6F 2B 73 75 n the mind to su
00000048: 65 66 65 72 0R 54 68 65 2B 73 6C 69 6E 67 73 2B ffer. The slings
00000058: 66 61 64 2B 61 67 72 72 6F 77 73 2B 6F 66 2B 73 and arrows of ou
00000068: 74 72 61 67 65 6F 75 73 2B 66 6F 72 74 75 66 65 tragoes fortune
00000078: 2C 0B 4F 72 2B 74 6F 2B 74 61 68 65 2B 61 72 66 , to take arm
00000088: 73 2B 61 67 61 69 6E 73 74 2B 61 2B 73 65 61 62 s against a sea
00000098: 6F 66 2B 74 72 6F 75 62 6C 65 73 2B 69 41 6E 64 of troubles. And
000000A8: 2B 62 70 74 78 70 74 69 65 68 65 69 65 68 65 64 by opposing end
000000B8: 34 68 65 60 69 54 6F 64 64 65 3B 64 65 3B 64 then; To die; t
000000C8: 6F 2B 6C 65 65 69 70 68 66 4E 6F 2B 60 6F 72 65 o sleep; No more
000000D8: 2B 2B 61 6E 64 2B 59 72 70 61 2B 73 6C 65 65 70 ; to sleep; a sleep
000000E8: 74 72 6F 78 73 61 79 2B 77 65 68 66 64 69 54 to say we end. T
000000F8: 69 65 2B 68 65 61 61 71 74 2D 61 63 69 65 2B 61 6E his heart-ache and
00000108: 80 69 64 65 61 69 71 74 2B 61 63 69 65 2B 61 6E d the thousand n
00000118: 64 2B 74 68 72 6F 75 73 61 6E 64 2B 6E atrual shucks. Th
00000128: 61 71 75 72 61 6C 73 68 65 63 6B 73 73 69 54 68 at flesh is heir
00000138: 61 74 2B 66 60 65 73 68 69 69 73 2B 69 65 69 72 at flesh is heir
00000148: 2B 74 6F 2C 2B 27 74 69 73 2B 61 2B 69 63 6F 6E 73 to , tis a cons
00000158: 75 6D 60 61 74 69 6F 66 84 44 65 76 76 75 74 6C ummaton. Devoutl
00000168: 79 2B 74 6F 2B 62 65 2B 77 69 73 68 72 27 64 2B 72 y to be wish'd.
00000178: 54 6F 2B 64 69 65 2C 2B 64 5B 72 73 6C 65 85 70 To die, to sleep
00000188: 3B 6B 54 6F 2B 73 6C 65 65 79 3B 2B 78 0B 72 63 ;To sleep; perc
00000198: 61 65 6E 63 65 2B 74 6F 2B 64 72 65 61 60 3B 2B chance to dream;
00000208: 61 79 2C 2B 74 68 65 72 65 27 2B 73 2B 74 68 65 2B there's the
00000218: 72 75 62 58 80 46 6F 72 20 69 6E 2B 74 68 61 74 rub; For in that
00000228: 2B 73 6C 65 65 70 2B 66 2B 64 65 61 74 68 20 sleep of death
00000238: 77 58 61 74 2B 64 72 65 61 60 73 2B 60 61 79 20 what dreams may
00000248: 63 6F 60 65 0B 57 68 65 6E 2B 77 65 2B 68 61 76 come. When we hav
```



hamlet-cipher.txt - Data

```
Long: $000005F0 | Type/Créateur: / | Sér: $00000380:$00000380:$00000000
00000000: 53 61 6C 74 65 65 64 5F 5F E9 1B C0 86 DC B1 6B C4 Salted_E.aU+^f
00000010: F7 R5 C8 F0 55 C5 CC 0F BF B3 C4 4F 54 2B 01 6F .o@n@.E@0!t+o
00000020: 43 13 60 B6 C3 R2 55 3C 56 E1 C3 53 6C 78 7F 4B C.m@r@U@P@*S!x.H
00000030: 2F 9E 98 5F 2D 4B 0B 12 BE F9 E9 87 EE FF 12 2B /08-@-8@.E@o..-
00000040: 36 6E 68 68 E0 D4 A0 0F F3 1F 6D F0 F7 E2 67 61K!n@t+0..m@+g
00000050: 66 91 10 81 FR 90 0B 0F 66 FR 66 67 4F 24 18 4D 0E fe,A u+ npG$6.
00000060: 85 6B 68 67 1A 91 FF 0B 0F 21 CH E2 27 FC 09 07 E7 K0ig,e@itL.vH
00000070: D8 F5 79 22 2B 29 16 0F 21 CH E2 18 7C C8 75 37 E7 r@i@y@.n@l@.-
00000080: 39 19 25 34 34 0F 07 01 3C 5C 55 C0 F4 90@.4@.7@.4@.5@.2@.0@.
00000090: 6B 66 62 2B 2D 2B 2B 9B 0E R1 FC E9 0B 0B 04 7C 0B 80 .o@n@.S@.1@.c
000000A0: 4D 6B 66 FF 84 R1 FC E9 0B 0B 04 7C 0B 80 .o@n@.S@.1@.c
000000B0: 58 R6 CC CD 4C B7 2B 22 94 F8 0B 0B 04 7C 0B 80 9C 21 15 X@Q@L@u@S@.5@!0!E
000000C0: 5B 60 89 01 R2 73 9E 81 3B C7 FC DC 3F 24 75 5A m!j.e@t@n@.c@.p@z@.
000000D0: 39 E6 R2 79 57 32 06 5F F9 88 1B 9D 3B F8 C6 D1 9E@y@2-@.d@!4-@-
000000E0: 29 F7 FF 87 62 R18 F7 59 01 2B 88 50 66 F2 62 )^@r@.v@.6!@b
000000F0: C9 4F 32 84 52 F2 72 R2 D8 DR 0R 93 2E FR 4E 54 .02R@N@u@p@/".%@.
00000100: FA 39 88 89 6F 6A 8E C8 66 2B 88 35 FB 80 59 3E 99@o@p@.f@.5@.c@.2@.
00000110: 93 ED AC 17 1D 92 6F 15 98 17 18 99 82 3B CD PR Al-.@.o@.6@!0
00000120: 98 5R 4C 5C 75 87 F1 11 94 2B 42 54 79 8C 74 2D .Z@L-@.s@.Bt@y@.
00000130: F3 33 5B 72 11 5B 1B 45 43 4F FC 28 8D AL 66 .3@r@.c@.ODD@.0@.
00000140: 59 2C 0B 78 72 2B 29 48 4E 19 58 0B F5 16 13 0D V,z@C@H@I.X,.i.,.
00000150: 7E 2D 27 33 37 98 CD 81 86 34 ED 29 32 43 39 61 .1@7@A@M@4@.2@9@.
00000160: 14 DF 0F 86 F4 8B E1 78 59 FR 22 13 D5 E8 F9 6B .f@o@y@x@.s@.E@h
00000170: 62 5E 2B 18 50 12 F1 RC E4 R4 E3 B3 88 66 74 F1 b^@.1@.E@k@.o@t@.
00000180: 91 BB 0B F1 4C 9B 0D C5 49 R6 E5 1B 0E 04 89 37 ar@R@V@P@.E@.7@
00000190: 4F C3 F9 67 94 8C 3B 58 04 CA F8 7D 60 BA DF 14 PO 00g@l@X..n@.f@.
00000200: 95 5E 07 F2 60 15 99 78 C0 09 C8 5B 5C 9F B@.A@.o@.V@.P@.
00000210: 0B 28 2B 0B F4 0D C2 B7 09 2B 80 88 2D BF CB FB .+,*@.2@.s@.r@.
00000220: Q9 72 5F 0F ER BD 91 02 3F CF 0A 2A 1C 7D E2 0F .0@.1@.o@.n@.p@.
00000230: ED 5D 1B 0B 38 84 CR C6 41 FF 70 3C DE EF 09 BE 11,8@.A@P@.n@p@e
00000240: C9 36 65 CC FE D3 0B 7C AR 7F D2 2A 3C 5A 5B P2 .6@.R,@.3@.*@.C@.
```

File encryption

File encryption with AES

Encryption

```
openssl enc -e -aes-256-cbc -k password -in hamlet.txt -out hamlet-cipher.txt
```

Decryption

```
openssl enc -d -aes-256-cbc -k password -in hamlet-cipher.txt
```

hamlet.txt - Data			
Long: \$000005D0	Type/Créateur:	/	Sér: \$00000000:00000000:\$00000000
00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			To be, or not to be, that is the question: Whether 'tis nobler i
00000010: 20 62 65 38 24 78 68 61 74 28 69 73 28 74 68 65			be; that is nobler in
00000018: 73 71 75 65 73 74 69 66 6E 0R 57 68 65 74 66			to be, or not to be; that is the question: Whether 'tis nobler i
00000028: 65 72 28 77 74 69 73 28 66 6F 62 6C 65 72 28 69			to be, or not to be; that is the question: Whether 'tis nobler i
00000038: 65 66 65 72 0R 54 68 65 20 73 6C 69 6E 67 73 28			in the mind to su
00000048: 65 66 65 72 0R 54 68 65 20 73 6C 69 6E 67 73 28			ffer. The slings
00000058: 65 66 65 72 0R 54 68 65 20 73 6C 69 6E 67 73 28			and arrows of ou
00000068: 74 72 61 67 68 6F 75 73 28 66 6F 72 74 75 66 65			trouges fortune
00000078: 2C 00 4F 72 28 74 6F 28 74 61 68 65 28 61 72 66			, to take arm
00000088: 73 26 61 67 61 69 6E 73 74 28 61 28 73 65 61 62			s against a sea
00000098: 6F 66 20 74 72 6F 75 62 6C 65 73 20 6R 41 6E 64			of troubles, And
000000A8: 20 62 79 70 78 68 69 65 66 67 68 69 65 66 64			by opposing end
000000B8: 34 68 65 66 67 68 69 54 6F 64 64 65 39 64			then; To die: t
000000C8: 6F 66 20 6C 65 65 70 39 68 4E 6F 20 60 6F 72 65			o sleep; Nor more
000000D8: 39 26 61 6E 64 29 62 79 26 61 29 73 6C 65 65 70			; to have a sleep
000000E8: 29 74 6F 68 73 61 79 28 77 65 68 66 64 69 54			; to say we end. T
000000F8: 69 65 26 68 65 61 71 74 20 61 63 69 65 26 61 6E			he heart-ache and
00000108: 64 28 74 68 65 29 74 68 6F 75 73 61 6E 64 29 6E			d the thousand n
00000128: 61 74 75 72 61 6C 73 68 6F 63 6B 73 98 54 68			atural shucks, Th
00000138: 61 74 20 66 60 65 73 68 69 73 28 66 65 69 72			at flesh is heir
00000148: 28 74 6F 2C 28 27 74 69 73 28 61 28 63 6F 6E 73			, to 'tis a cons
00000158: 75 6D 60 61 74 69 6F 66 8A 44 65 76 75 74 6C			ummation. Devoutl
00000168: 79 28 74 6F 28 62 65 28 77 69 73 68 27 64 2E 2B			y to be wish'd.
00000178: 54 6F 28 64 69 65 2C 28 64 28 73 6C 65 65 70			To die, to sleep
00000188: 38 66 54 6F 28 73 6C 65 65 70 38 28 78 05 72 63			; To sleep: perh
00000198: 61 65 6E 63 28 74 6F 28 64 72 65 61 60 38 20			hance to dream;
000001A8: 61 79 2C 20 74 68 65 72 65 27 23 78 74 68 65 20			there's the rub; In that
000001B8: 72 75 62 58 08 46 6F 72 20 69 6E 20 74 68 61 74			rub; For in that
000001C8: 20 73 6C 65 65 70 20 66 28 64 65 61 74 68 20			sleep of death
000001D8: 77 68 61 74 20 64 72 65 61 60 73 28 60 61 79 20			what dreams may
000001E8: 63 6F 60 65 08 57 68 65 6E 20 77 65 28 68 61 76			come. When we hav

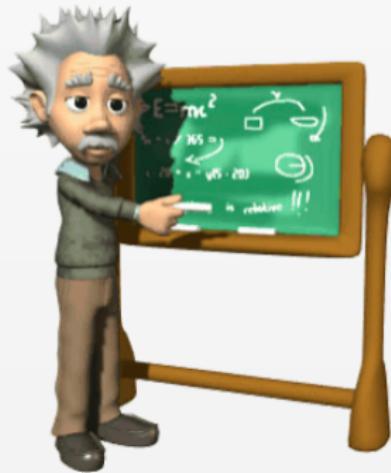
hamlet-cipher.txt - Data			
Long: \$000005F0	Type/Créateur:	/	Sér: \$00000380:00000380:\$00000000
00000000: 53 61 6C 74 65 6F 5F E9 1B C0 86 DC B1 68 C4			Saltsed_E.aU+^f
00000010: 7F R5 C8 F0 55 C5 CC 5C 5F BF B3 C4 4F 54 2B 01 6F			.onM+eVp*+S1x.H
00000020: 43 13 60 B6 C3 R2 55 5C 5E E1 C3 53 6C 78 7F 48			C.m+eVp*+S1x.H
00000030: 2F 9E 98 5F 20 40 08 12 BE F9 E9 87 EE FF 12 2E			/08-8-E^Ed0..
00000040: 36 6E 68 68 E0 D4 08 F3 1F 6D F0 F7 E2 67 61KtH0+1.0.m+eVg			..9
00000050: 66 91 10 81 FR 90 08 0F 66 FR 6E 67 4F 24 18 40 0E			fe.A.u+npG\$6.
00000060: 68 85 EB 61 1A 91 FF 09 08 27 FC 09 07 ET K0ig.e+itL.vH			koig.e+itL.vH
00000070: 79 22 28 16 0F 21 CH E2 18 7C C8 75 37 07			7.riy+u.f+nfJ7.
00000080: 39 19 25 08 34 8F 07 3C 98 B4 55 08 C0 F4 900148.../45e520			900148.../45e520
00000090: 60 68 62 21 20 9B 9E A1 FC E9 09 08 04 7C 80 80			0b53...92S.154+-c
000000A0: 4D 68 62 21 20 9B 9E A1 FC E9 09 08 04 7C 80 80			0b53...92S.154+-c
000000B0: 58 R6 CC CD 4C B7 2R 22 94 08 FB F4 09 90 21 15			X500148...\$1501E
000000C0: 58 R6 CC CD 4C B7 2R 22 94 08 FB F4 09 90 21 15			X500148...\$1501E
000000D0: 39 E6 R2 79 57 32 06 5F F9 88 18 90 38 F8 C6 D1 9EpujL2-3.0.014-			m1.e+ut+e.../7uZ
000000E0: 29 F7 FF 87 62 R18 F2 79 57 09 81 28 88 50 66 F2 32)^+a+e..v. 6)z>b)^+a+e..v. 6)z>b
000000F0: C9 AF 32 84 52 F2 72 R2 D8 DR 0R 93 2E FR 4E 64 .02RUnU+g/".8%			.02RUnU+g/".8%
00000100: FR 39 88 B8 6F 68 E8 C8 66 2F B8 35 FB 80 59 36			9090+jewf./5*c22
00000110: 98 ED AC 17 1D 92 6F 15 98 17 18 99 82 3B CD PR Al...i...6c10			Al...i...6c10
00000120: 09 5C 45 C5 87 F1 18 94 26 RB 42 54 79 8C 74 2D			-ZL=2...-s.Btuht
00000130: F3 33 58 72 11 58 1B 45 43 4F BC 28 BD AL 66 .3^r...1...OD03(0^F			
00000140: 59 2C 08 78 72 28 29 48 4E 19 58 08 F5 16 13 0D Vz.C7H1N..i..E			
00000150: 7F ED 27 33 37 98 CD 81 86 34 ED 29 32 43 39 61 .1'37a04u(4)209a			
00000160: 14 DF 6F 86 F8 E1 78 59 FR 22 13 D5 E8 F9 68 .fcoju...xx...-E'h			
00000170: 62 5E 28 18 50 12 F1 RC E4 E8 B3 88 66 74 F1 b^+1.0.E8g+af0t0			
00000180: 91 BB 08 F1 RC 9D C5 49 E8 E6 1B 0E 04 89 37 erQ...V@P...E..?7			
00000190: 4F C3 F9 67 94 8C 58 58 04 C8 F7 60 BA DF 14 RD 00gjg@X..n...f#			
000001A0: 95 5E 07 F2 60 15 99 78 C0 98 C9 5C 59 C0 F8 ^...u...d...V...n...p...			
000001B0: 08 28 80 FB 04 DD C2 B7 9C 28 AD 80 20 CB FB +...+...-2...+E+R...@			
000001C0: Q9 72 PF 0E RD 01 23 F9 CF 0A 21 1C 7D E2 07 F9 .0...10...7...+...+.../			
000001D0: ED 5D 1B 08 38 84 CR C6 41 FF 70 3C DE EF 09 BE 11,8R AFp+n09e			
000001E0: C9 36 65 CC FE D3 08 7C AR 7F D2 2A 3C 5A 5B P2 .6...R...[3...**_C e			

In practice

Tutorial: CrypTool

- ▶ Download and install CrypTool
- ▶ http://www.cryptool.org/download/SetupCryptool_1_4_30_en.exe
- ▶ With the aid of the project documentation use this toolbox to play with cryptography
- ▶ Have fun ;-)

Questions ?



Bibliography



[Portal:Cryptography](#)

Wikipedia Fundation

<http://en.wikipedia.org/wiki/Portal:Cryptography>



[Sage Reference – Cryptography](#)

Sage Project

<http://www.sagemath.org/doc/reference/cryptography.html>



[Wolfram – Mathworld](#)

Alfred Menezes, Paul van Oorschot and Scott Vanstone

<http://www.cacr.math.uwaterloo.ca/hac/>



[Handbook of Applied Cryptography](#)

Eric Weisstein

<http://mathworld.wolfram.com/>

Bibliography

-  **Applied Cryptography**
Bruce Schneier
<http://www.schneier.com/book-applied.html>
-  **The Codebreakers**
David Kahn
<http://www.david-kahn.com>
-  **Cryptool**
Community of contributors
<http://www.cryptool.org/>

Licence

You are free to:

- ▶ share: copy, distribute and transmit this work
- ▶ remix: adapt this work

Attribution



You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Non commercial



You may not use this work for commercial purposes.

Share Alike



If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.