isep
École d'ingénieurs du numérique

Cybersecurity Course

Lecture 1: Introduction to Cryptography

Nour EL MADHOUN

Associate Professor

nour.el-madhoun@isep.fr

Office: L219

Cybersecurity aims to defend

Computers

Servers

Data

Smartphones

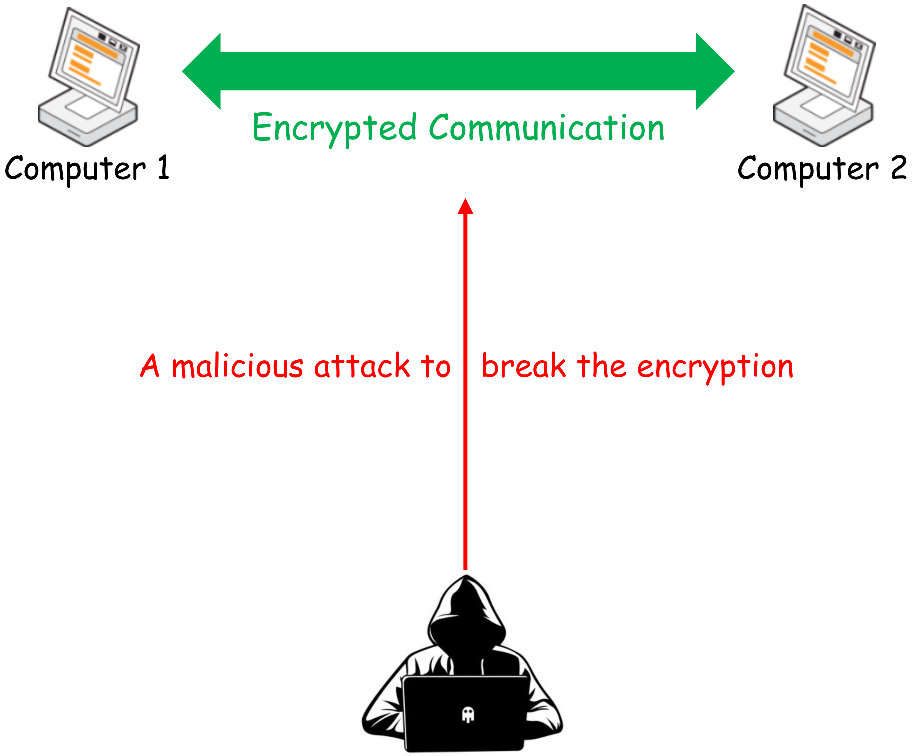
Network

against malicious attacks

What is an attack ?

* It is a malicious action that does not comply with the security policy of a system

Example



Computer 1

Encrypted Communication

Computer 2

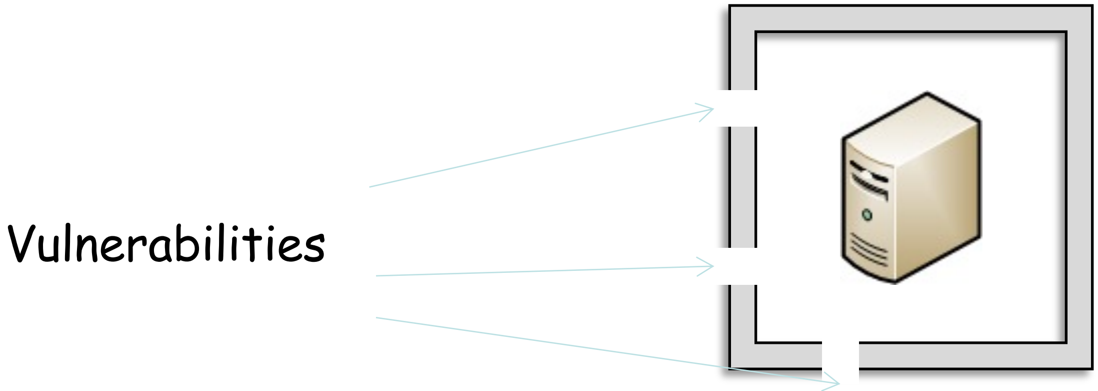
A malicious attack to break the encryption

What is an attack ?

* It is also the exploitation of a vulnerability

What is a vulnerability ?

* It is a weakness in the system that can be exploited for unintended purposes

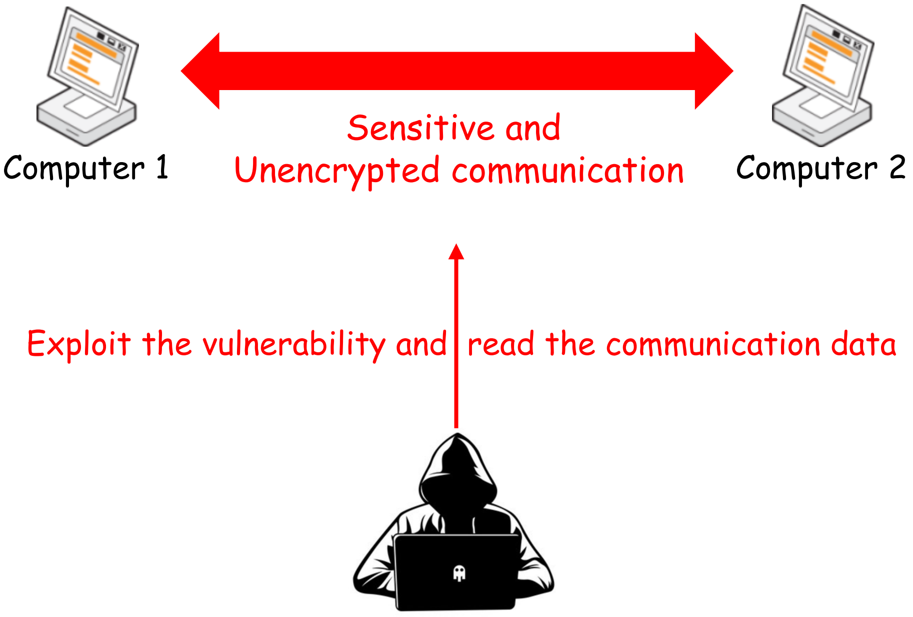


What is an attack ?

* It is also the exploitation of a vulnerability

What is a vulnerability ?

* Example: the lack of guaranteeing the confidentiality of exchanges in a sensitive communication



Computer 1

Sensitive and
Unencrypted communication

Computer 2

Exploit the vulnerability and read the communication data

What are the motivations of an attacker?

- Money, personal reasons

- Political commitment

- The Grudge

- Curiosity ….

- Stupidity! To impress friends ... etc.

Cyberattacks: Examples

Cyberattacks : Examples

- 50 million Facebook Accounts hacked, in September 2018:
  - Security vulnerability in the app's code
- Airbus - January 2019
  - Cyberattack on company data: business details, employee identities were accessed by hackers

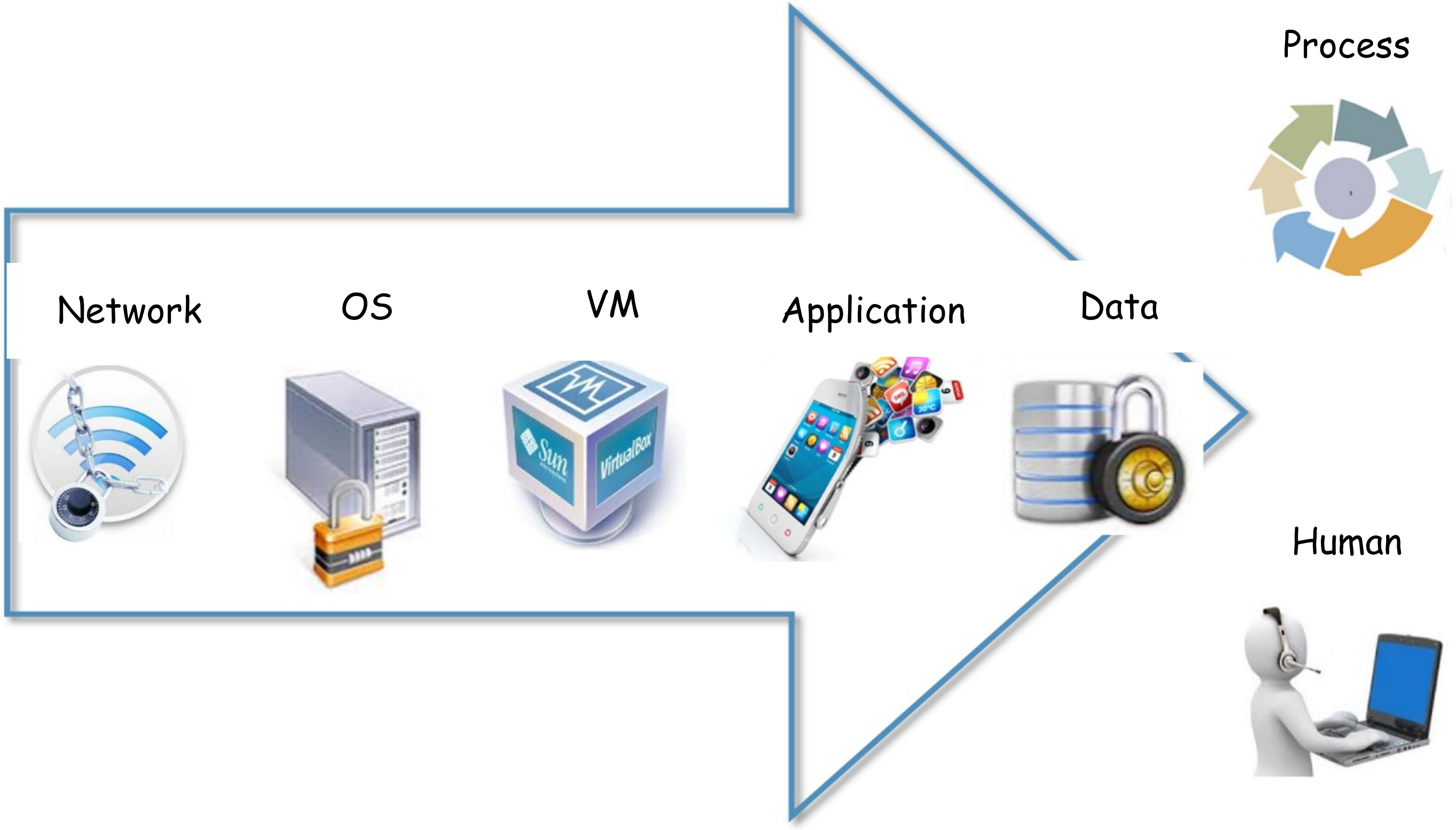
Cyberattacks : Examples

- CHU (Centre Hospitalier Universitaire) of Rouen - November 2019
  - The hacker blocked the machines and demanded a ransom to restart them
  - The attack caused a general shutdown of equipment affecting IT, elevators, medical imaging, . .

Targets

- Banks
- Military servers
- Universities
- Internet service providers
- Networks, communications, etc.

Cybersecurity is at all levels



Network

OS

VM

Application

Data

Process

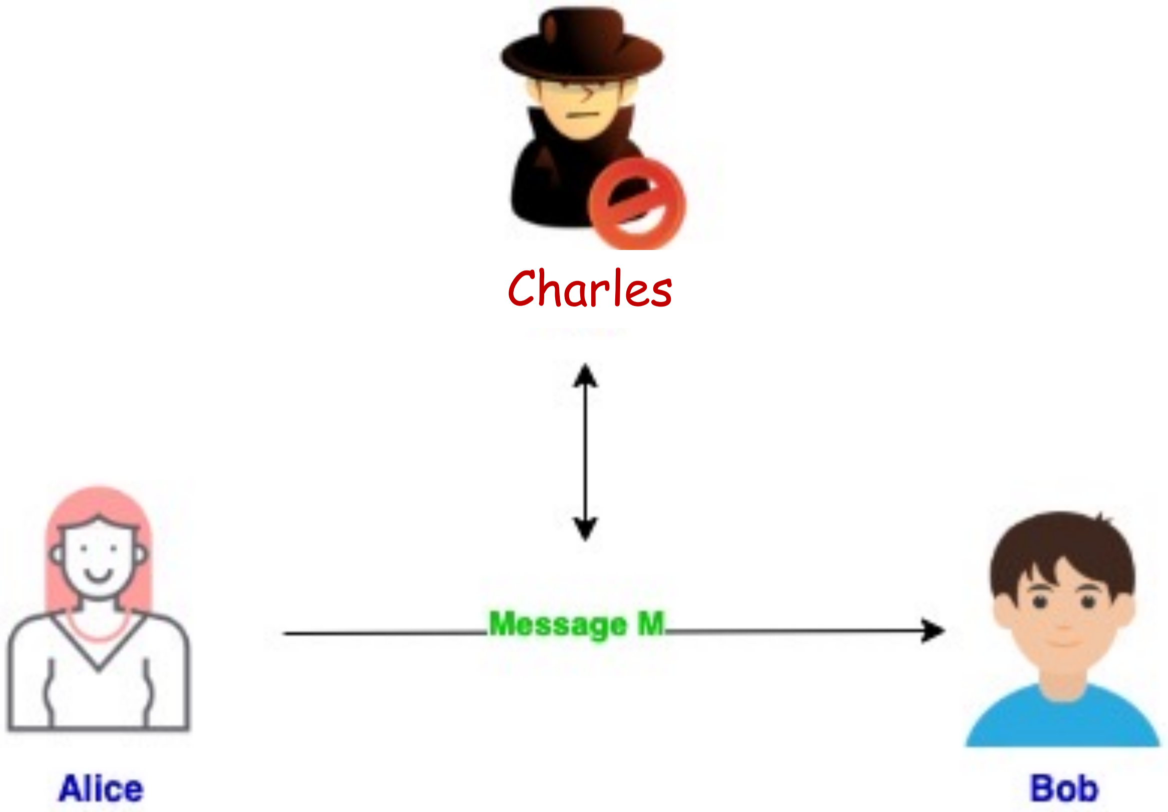
Human

How can cybersecurity make a system more secure ?

4 security properties are used !

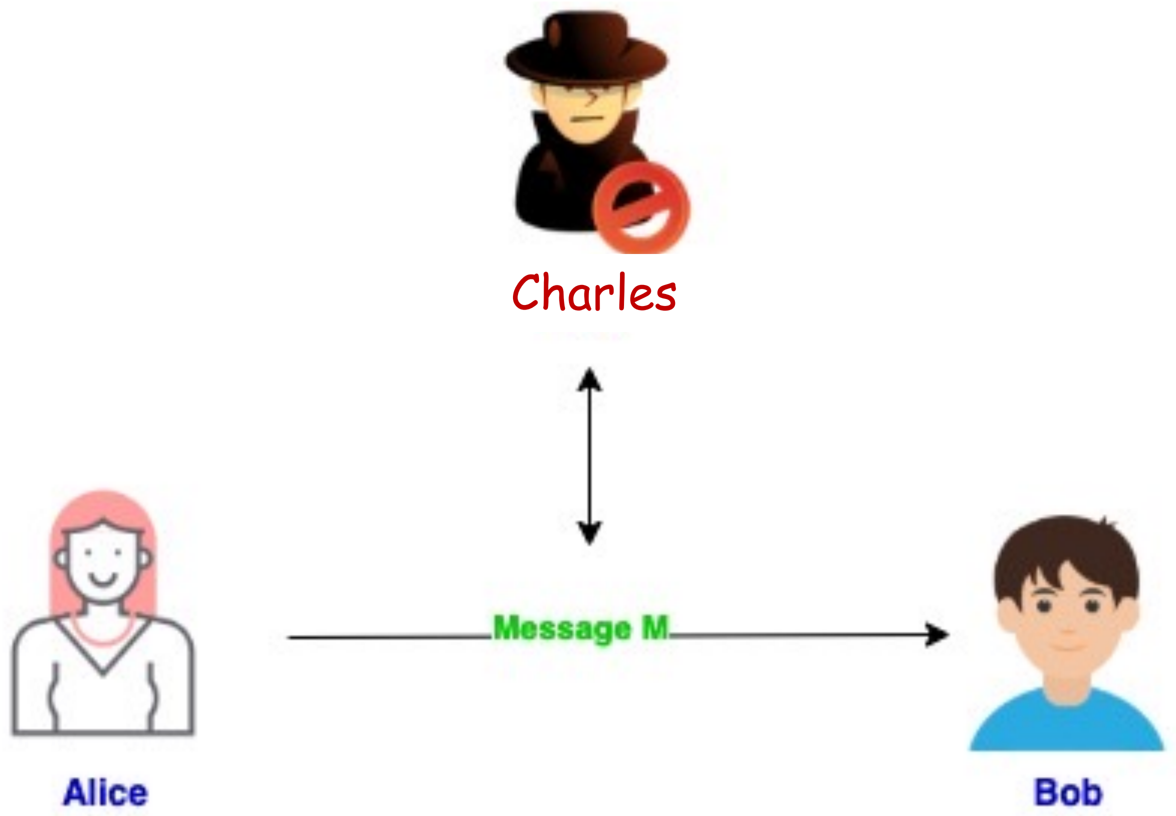
Confidentiality Integrity Authentication Non-repudiation

Confidentiality



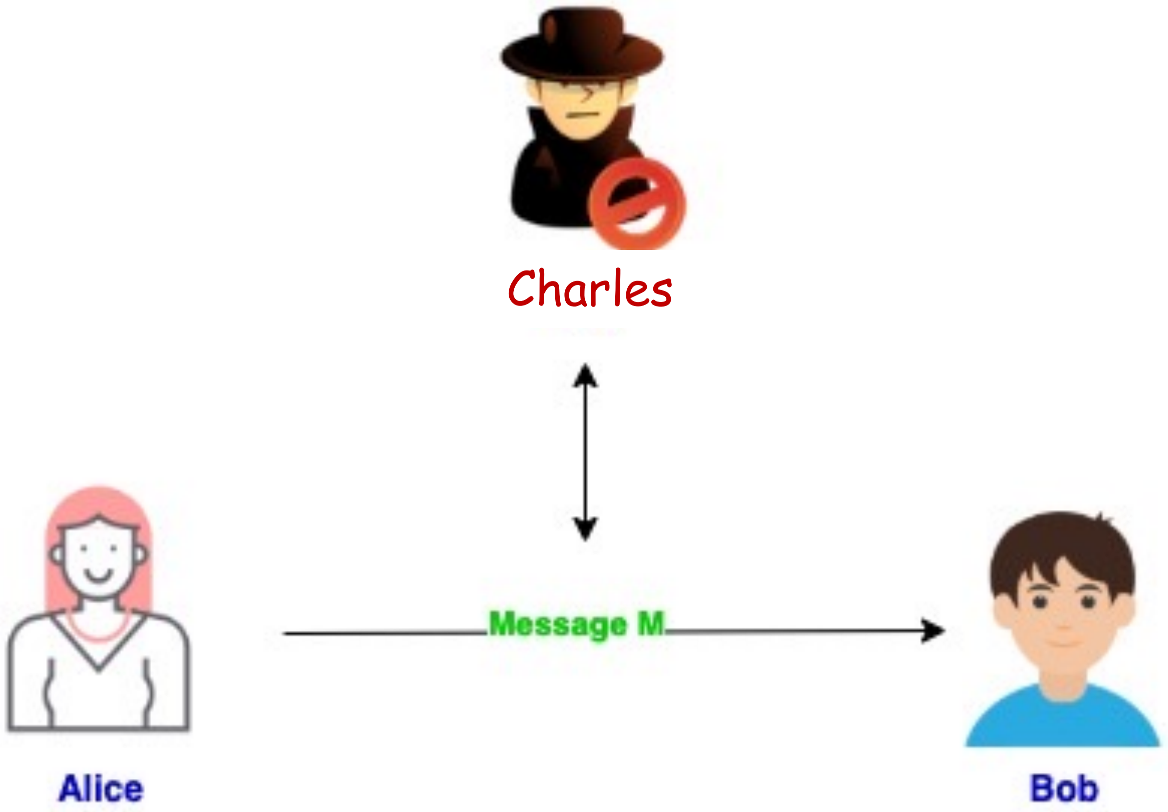
- ✓ Prevent Charles from intercepting and reading the contents of the M
- ✓ Only Alice and Bob who can understand the contents of the M

Integrity



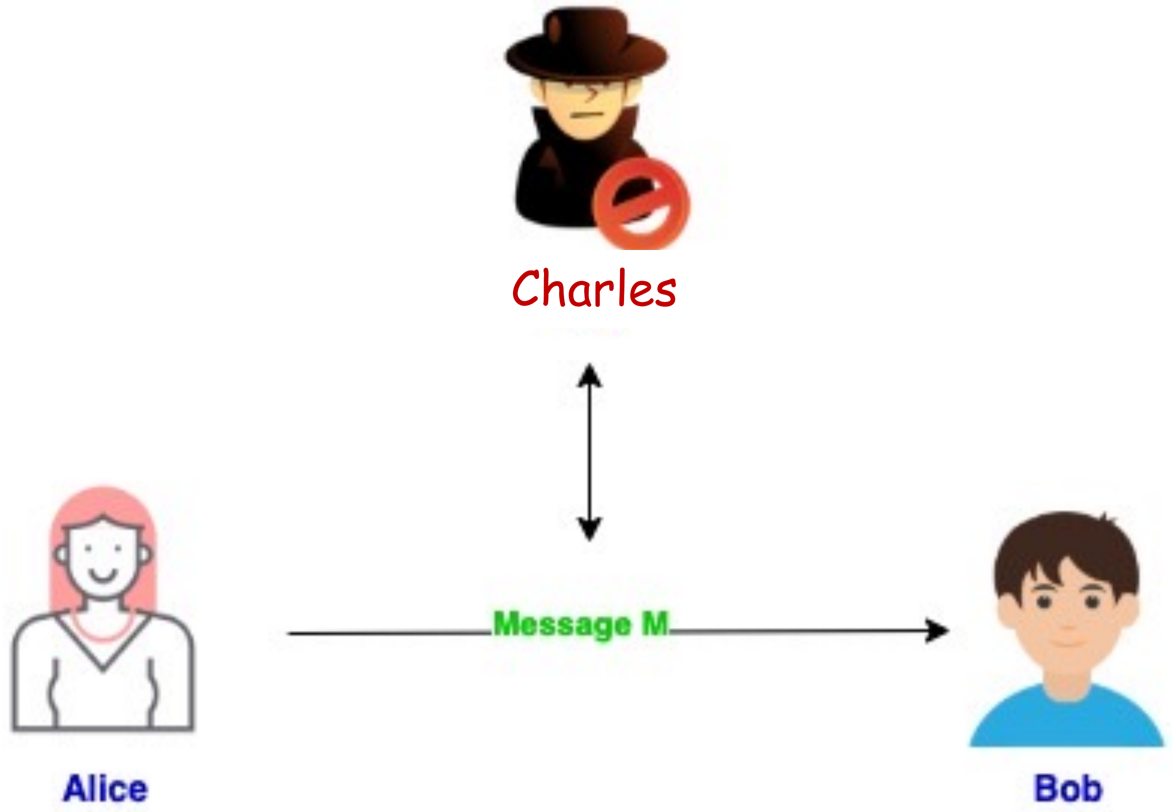
- ✓ Prevent Charles from changing the content of the M
- ✓ The content of the M is not modified, maliciously or accidentally, during transmission

Authentication



- ✓ Prevent Charles from taking the identity of Alice or Bob
- ✓ Alice must be authenticated to Bob
- ✓ Bob must be authenticated to Alice

Non-repudiation

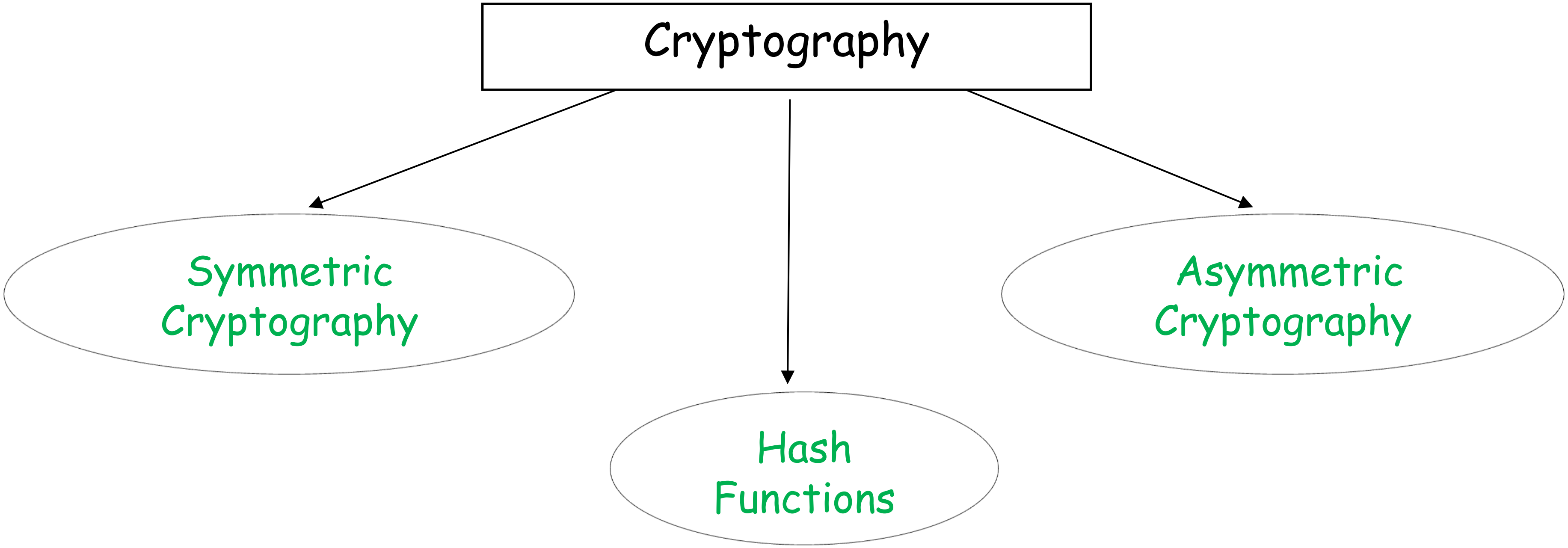


- ✓ Alice can not deny that she sent M

Question:

How are these properties ensured?

Answer: They are ensured thanks to the Cryptography !



Cryptography

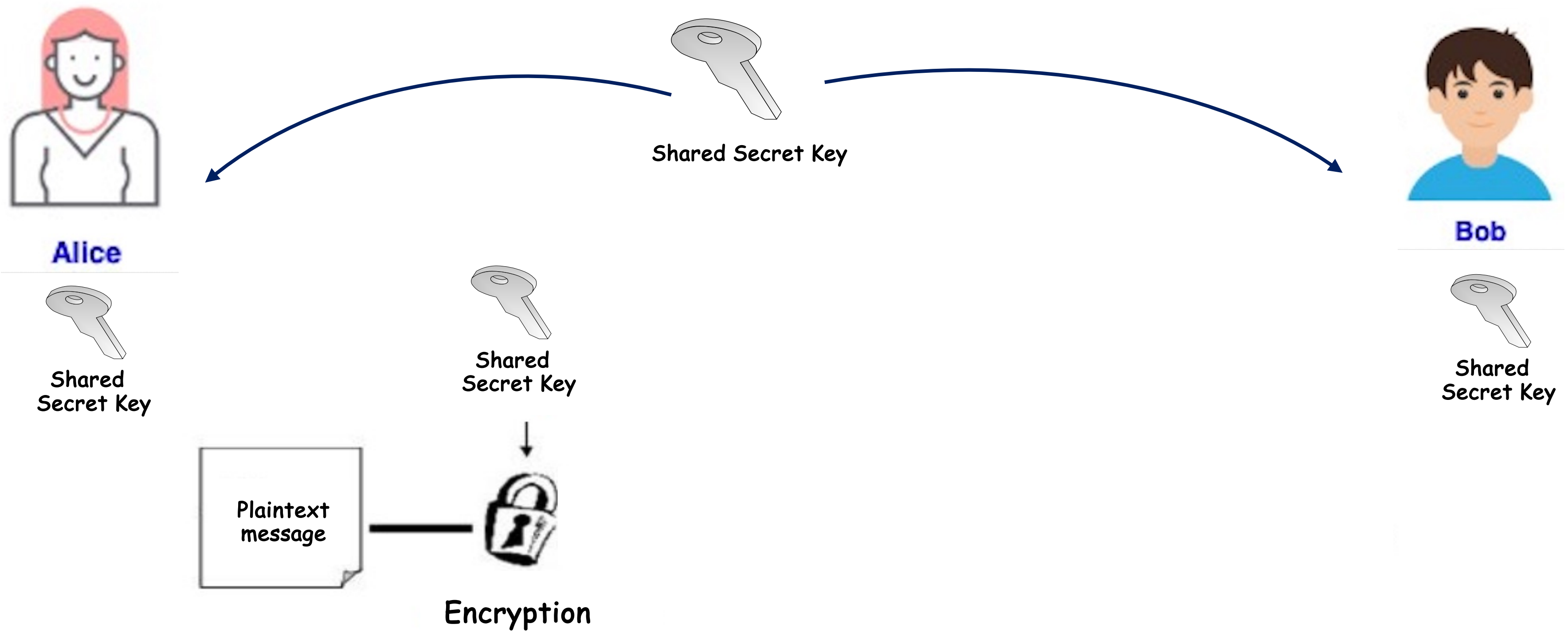
Symmetric Cryptography

Hash Functions

Asymmetric Cryptography

Cryptography

Symmetric Cryptography



Shared Secret Key

Alice

Shared Secret Key

Shared Secret Key

Bob

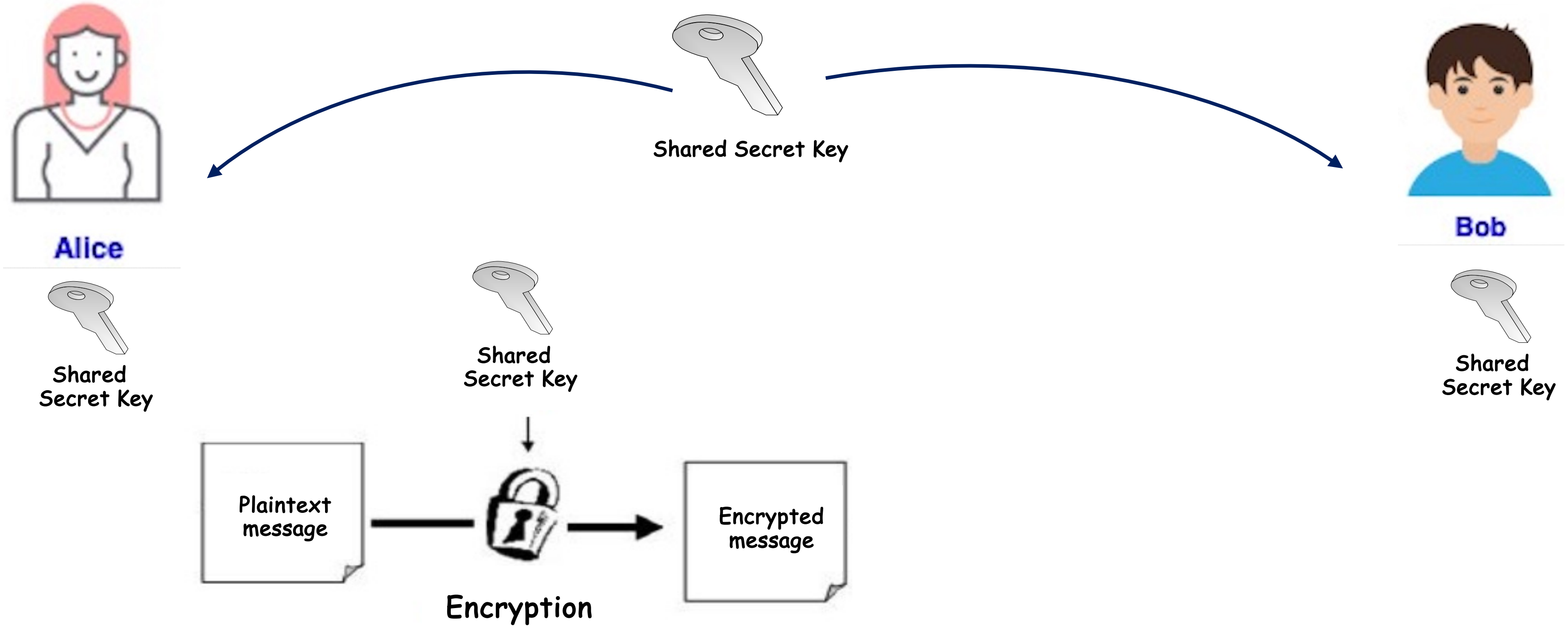
Shared Secret Key

Plaintext message

Encryption

Cryptography

Symmetric Cryptography



Shared Secret Key

Alice

Shared Secret Key

Shared Secret Key

Bob

Shared Secret Key

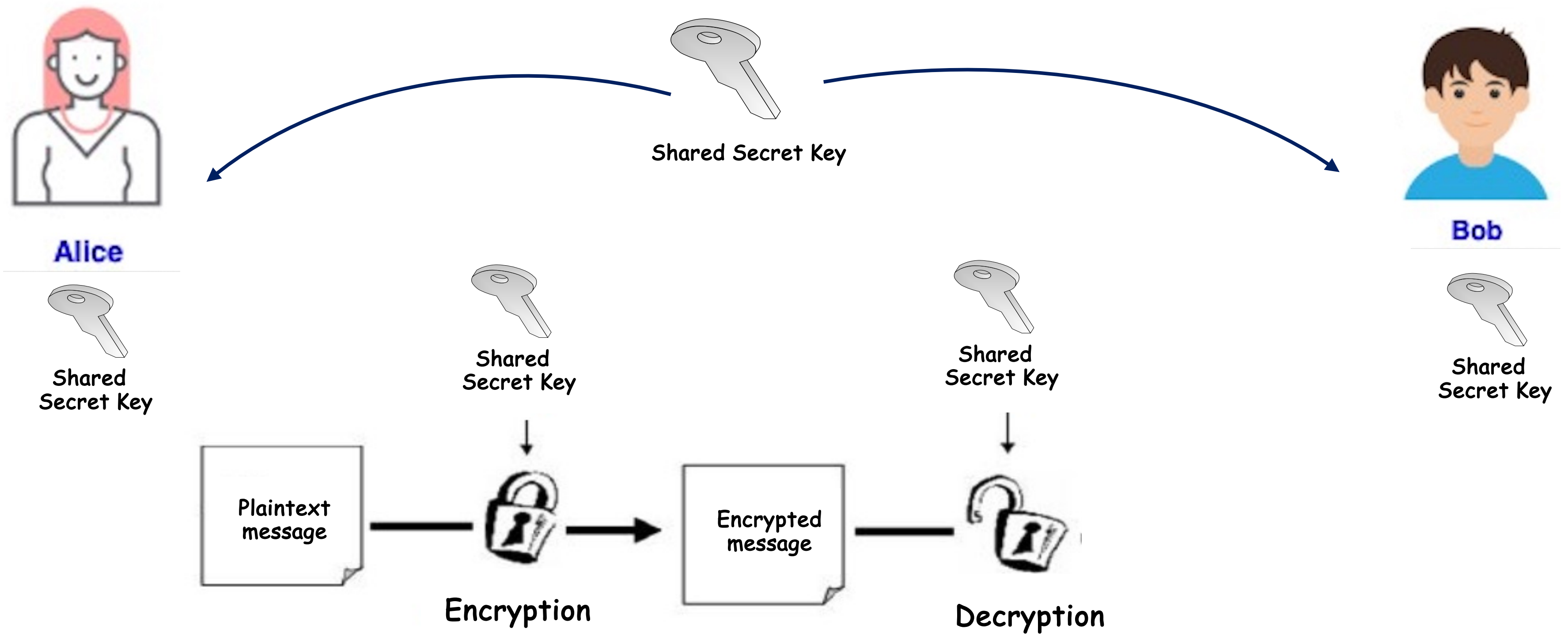
Plaintext message

Encryption

Encrypted message

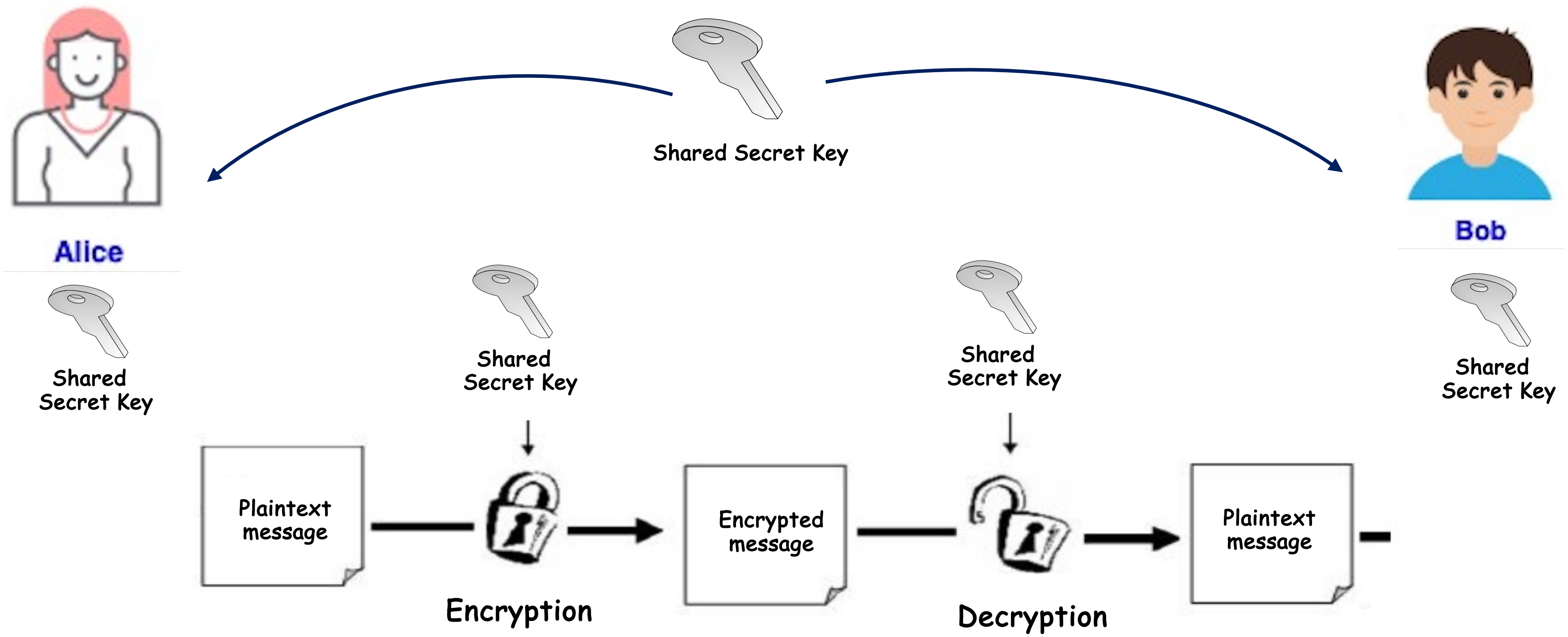
Cryptography

Symmetric Cryptography



Cryptography

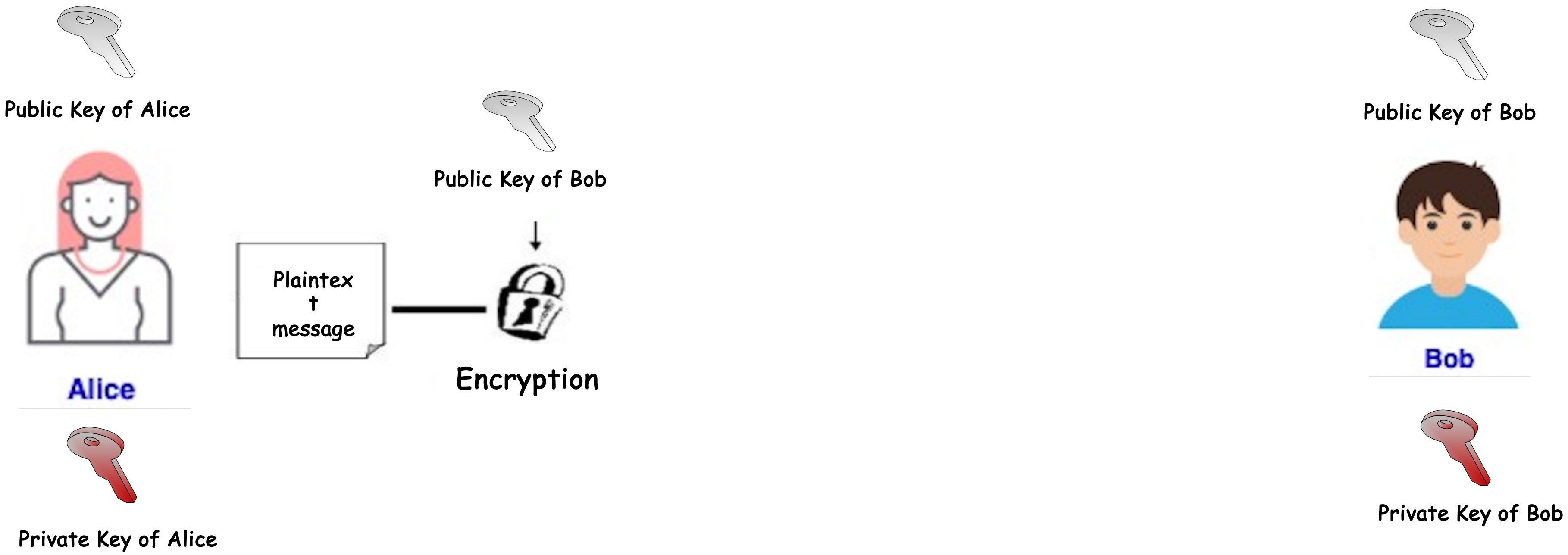
Symmetric Cryptography



Confidentiality of the message

Cryptography

Asymmetric Cryptography



Public Key of Alice

Alice

Private Key of Alice

Plaintext message

Public Key of Bob

Encryption

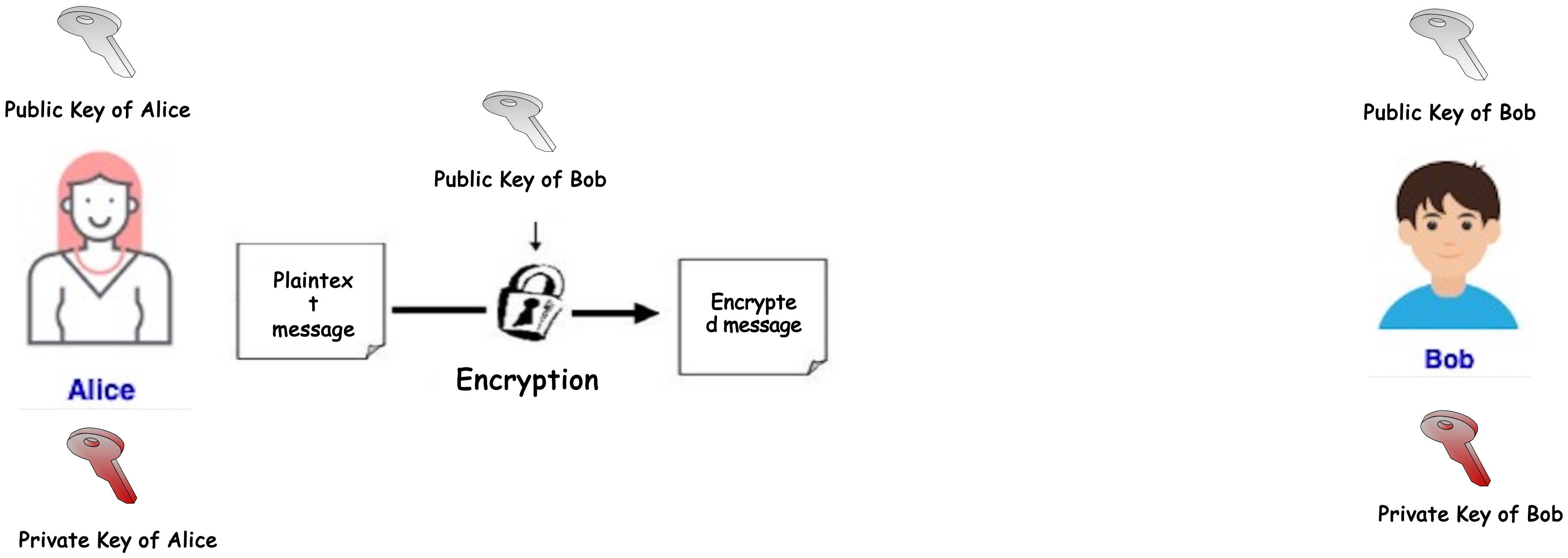
Public Key of Bob

Bob

Private Key of Bob

Cryptography

Asymmetric Cryptography



Public Key of Alice

Alice

Private Key of Alice

Plaintex
t
message

Public Key of Bob

Encryption

Encrypte
d message

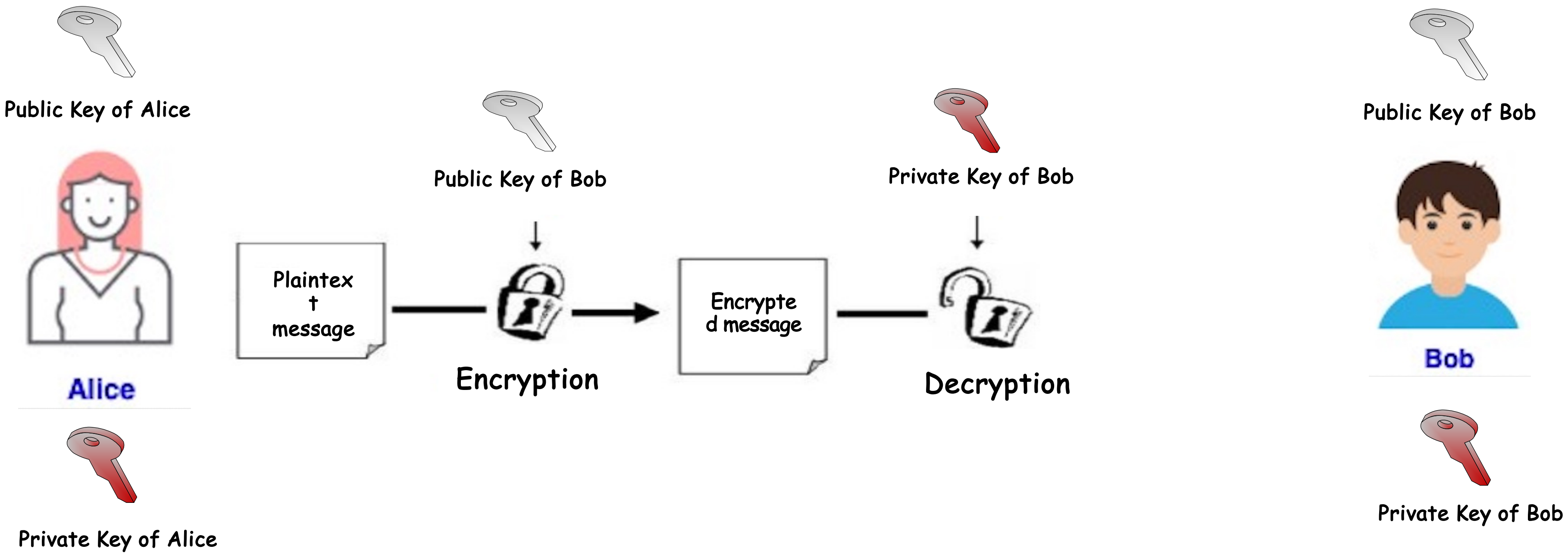
Public Key of Bob

Bob

Private Key of Bob

Cryptography

Asymmetric Cryptography



Public Key of Alice

Alice

Private Key of Alice

Plaintext message

Public Key of Bob

Encryption

Encrypted message

Private Key of Bob

Decryption

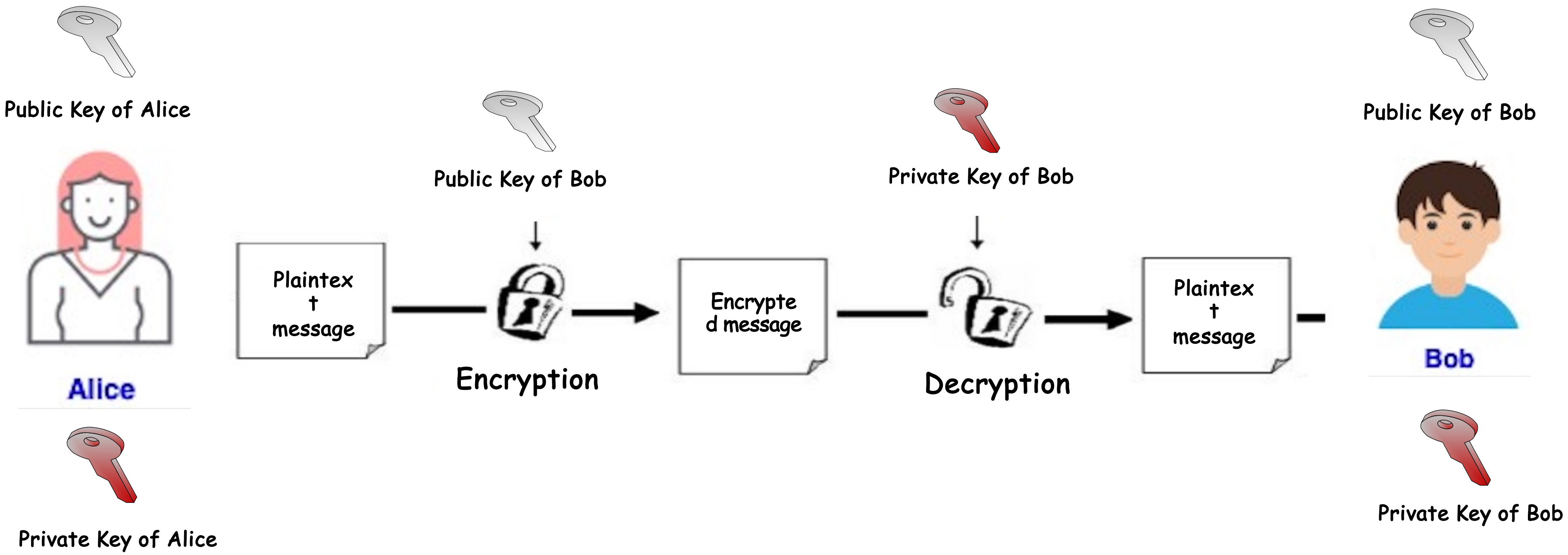
Public Key of Bob

Bob

Private Key of Bob

Cryptography

Asymmetric Cryptography



Public Key of Alice

Alice

Private Key of Alice

Plaintext message

Public Key of Bob

Encryption

Encrypted message

Private Key of Bob

Decryption

Plaintext message

Public Key of Bob

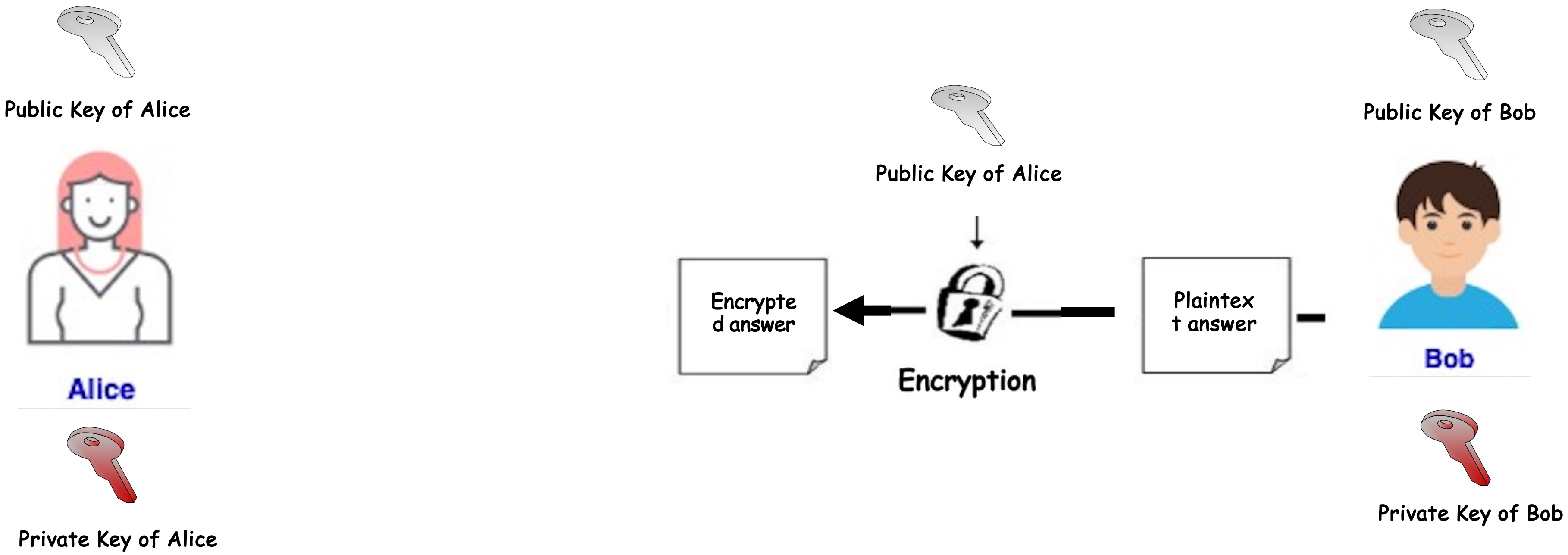
Bob

Private Key of Bob

Confidentiality of the message

Cryptography

Asymmetric Cryptography



Public Key of Alice

Alice

Private Key of Alice

Public Key of Alice

Encrypte d answer

Encryption

Plaintex t answer

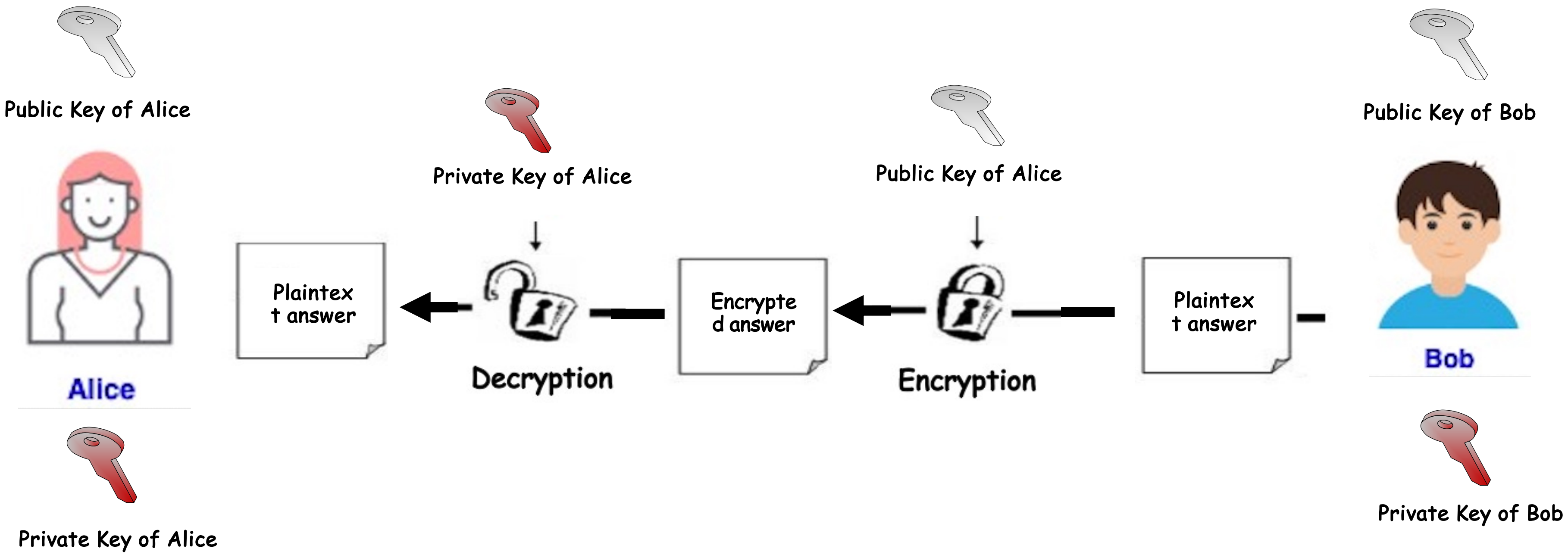
Public Key of Bob

Bob

Private Key of Bob

Cryptography

Asymmetric Cryptography



Public Key of Alice

Alice

Private Key of Alice

Plaintext answer

Private Key of Alice

Decryption

Encrypted answer

Public Key of Alice

Encryption

Plaintext answer

Public Key of Bob

Bob

Private Key of Bob

Confidentiality of the answer

Cryptography

Symmetric Cryptography → Confidentiality of the message

Asymmetric Cryptography → Confidentiality of the message

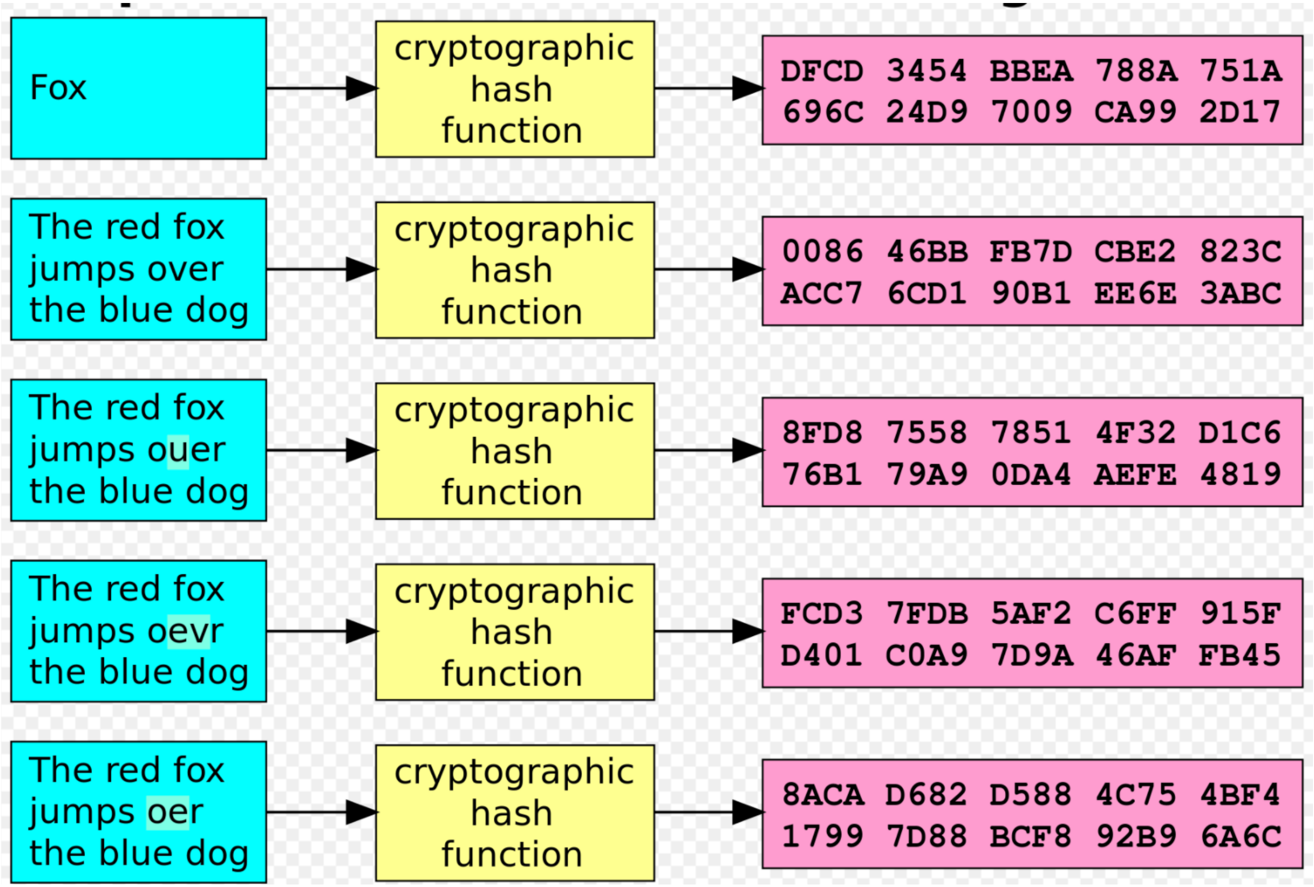
Hash Functions → Integrity of the message

Cryptography

Hash Functions

Input (Message)

Output (Hash)



M1

Fox → cryptographic hash function → DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17

H1: Fixed size

M2

The red fox jumps over the blue dog → cryptographic hash function → 0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC

H2: Fixed size

M3

The red fox jumps ouer the blue dog → cryptographic hash function → 8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819

H3: Fixed size

M4

The red fox jumps oevr the blue dog → cryptographic hash function → FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45

H4: Fixed size

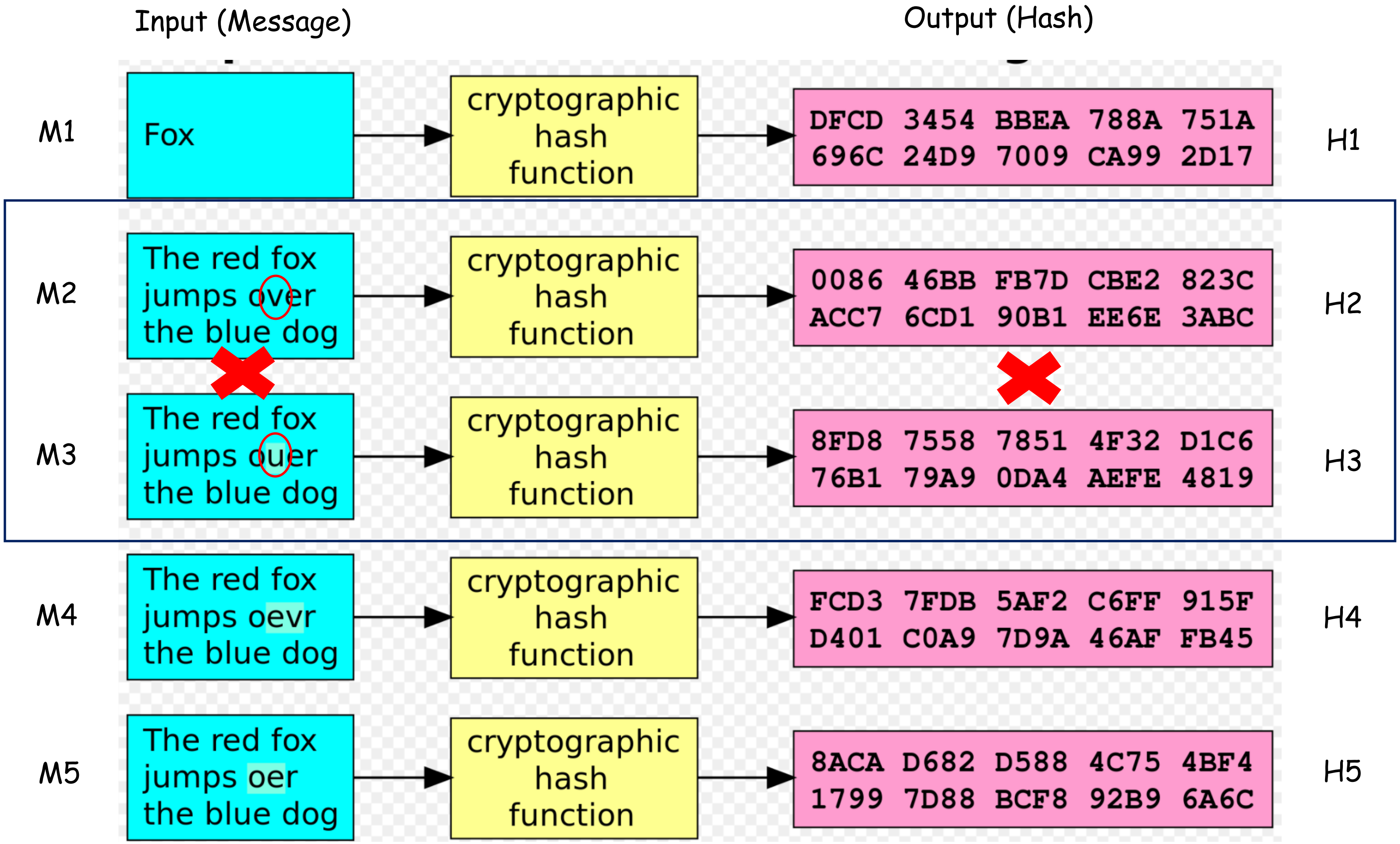
M5

The red fox jumps oer the blue dog → cryptographic hash function → 8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

H5: Fixed size

Cryptography

Hash Functions



Input (Message)

Output (Hash)

M1 — Fox → cryptographic hash function → DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17 — H1

M2 — The red fox jumps over the blue dog → cryptographic hash function → 0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC — H2

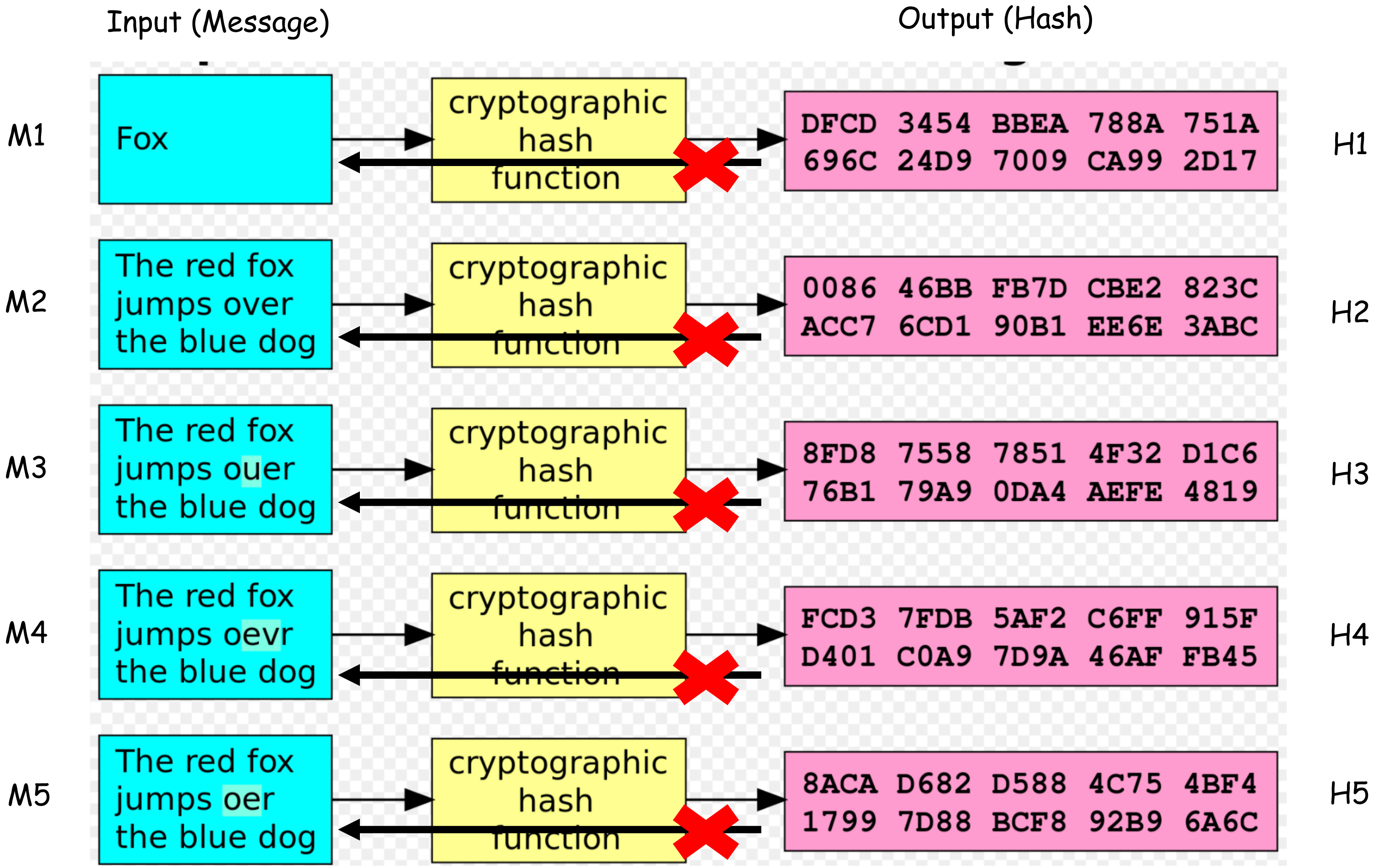
M3 — The red fox jumps ouer the blue dog → cryptographic hash function → 8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819 — H3

M4 — The red fox jumps oevr the blue dog → cryptographic hash function → FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45 — H4

M5 — The red fox jumps oer the blue dog → cryptographic hash function → 8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C — H5

Cryptography

Hash Functions



Input (Message)

Output (Hash)

M1 Fox → cryptographic hash function → DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17 H1

M2 The red fox jumps over the blue dog → cryptographic hash function → 0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC H2

M3 The red fox jumps ouer the blue dog → cryptographic hash function → 8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819 H3

M4 The red fox jumps oevr the blue dog → cryptographic hash function → FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45 H4

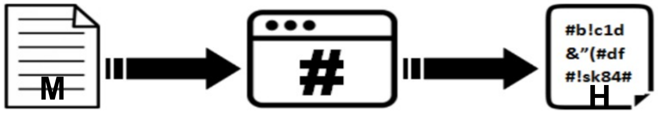
M5 The red fox jumps oer the blue dog → cryptographic hash function → 8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C H5

Cryptography

Hash Functions

Alice

Bob



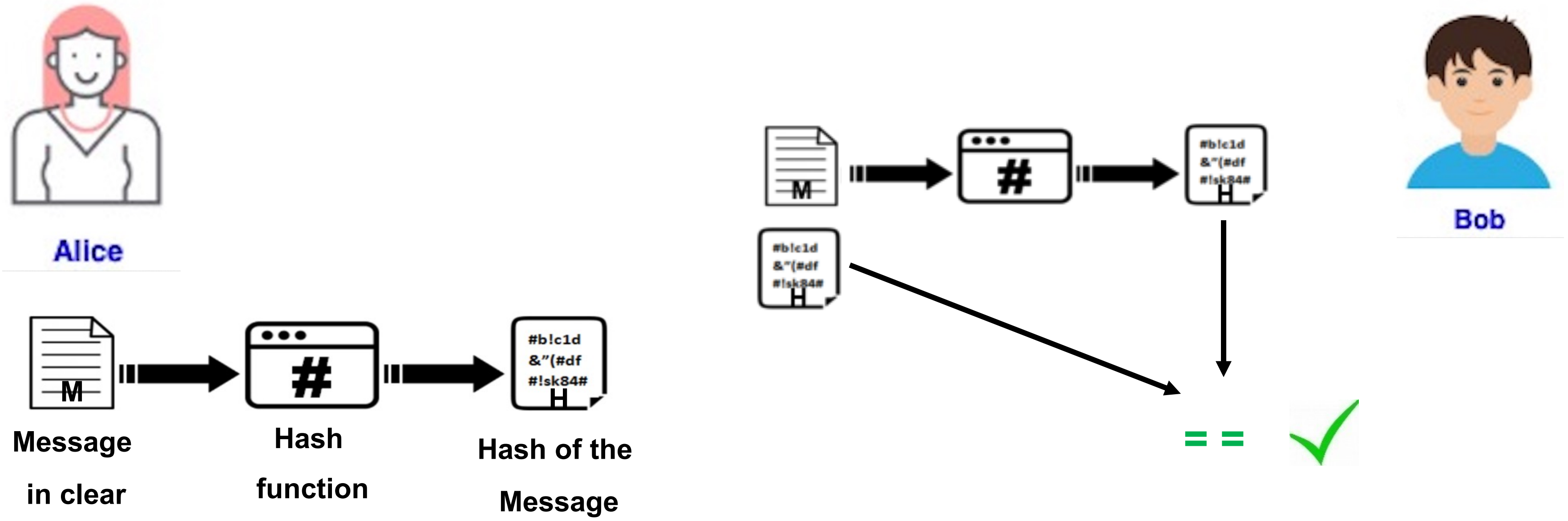
Message in clear

Hash function

Hash of the Message

Cryptography

Hash Functions



Alice

Message in clear

Hash function

Hash of the Message

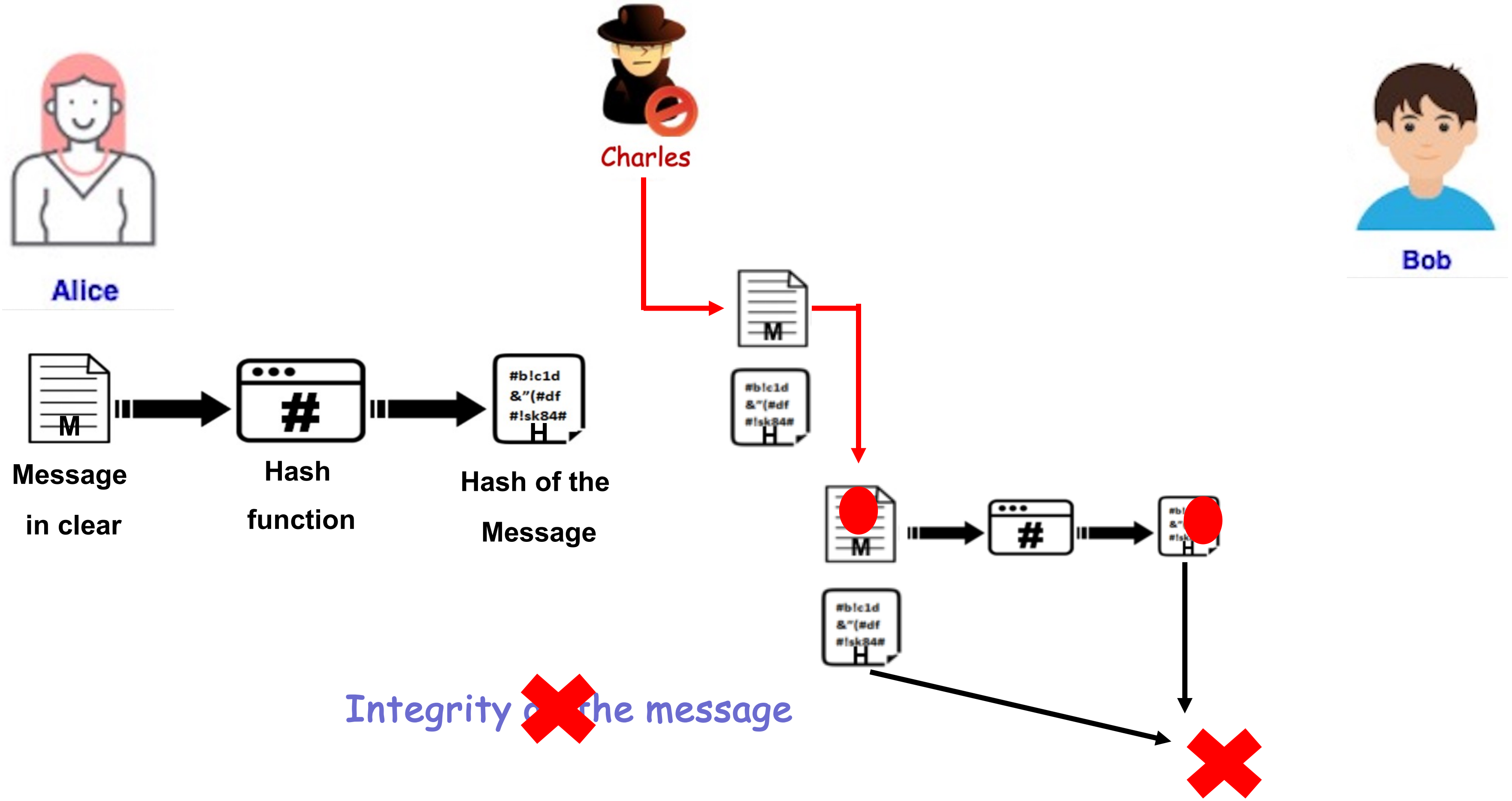
Bob

= =

Integrity of the message

Cryptography

Hash Functions



Integrity ✖ of the message

Cryptography

Symmetric Cryptography → Confidentiality of the message

Asymmetric Cryptography → Confidentiality of the message

Hash Functions → Integrity of the message

Authentication and Non-repudiation ?

Asymmetric Cryptography
+
Hash Function

Cryptography

Authentication and Non-repudiation ?

Asymmetric Cryptography

+

Hash Function

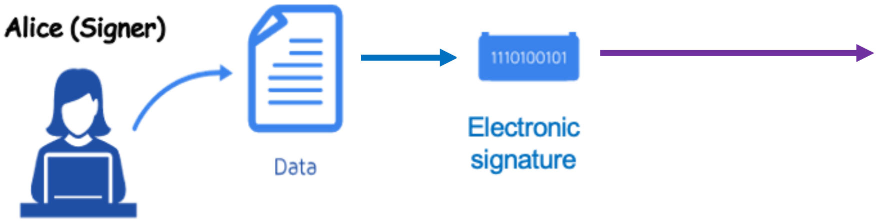
Electronic Signature

Cryptography

Authentication and Non-repudiation ?

Asymmetric Cryptography
+
Hash Function

Electronic Signature



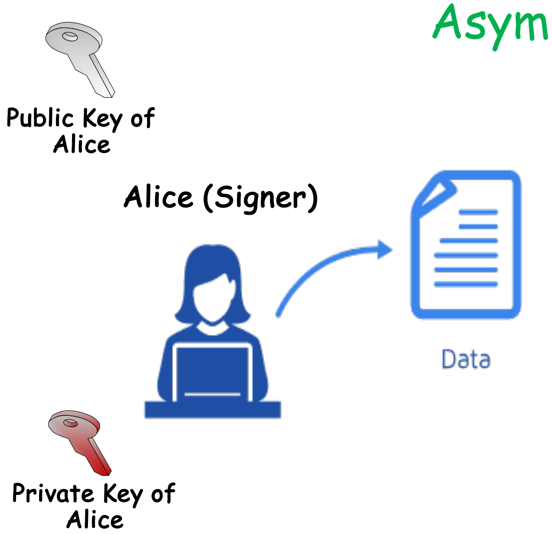
Authentication of the Signer

Non-repudiation for the Signer

Integrity of the Data

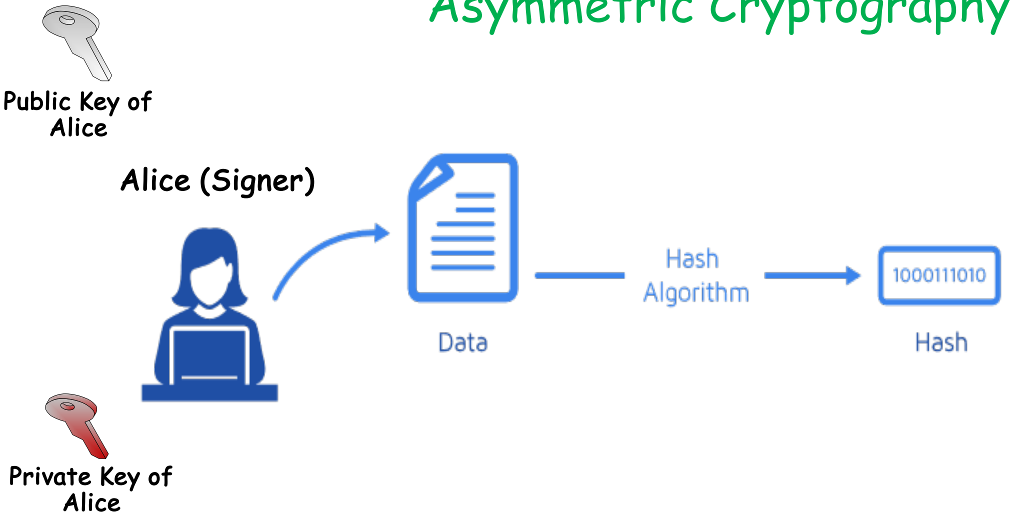
Electronic Signature

Asymmetric Cryptography + Hash Function



Electronic Signature

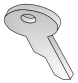
Asymmetric Cryptography + Hash Function



Bob (Receiver)

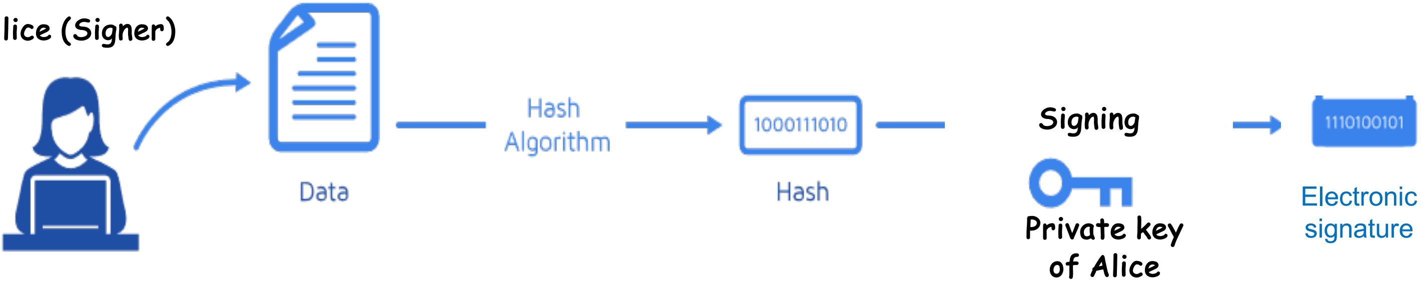
Electronic Signature

Asymmetric Cryptography + Hash Function



Public Key of Alice

Alice (Signer)

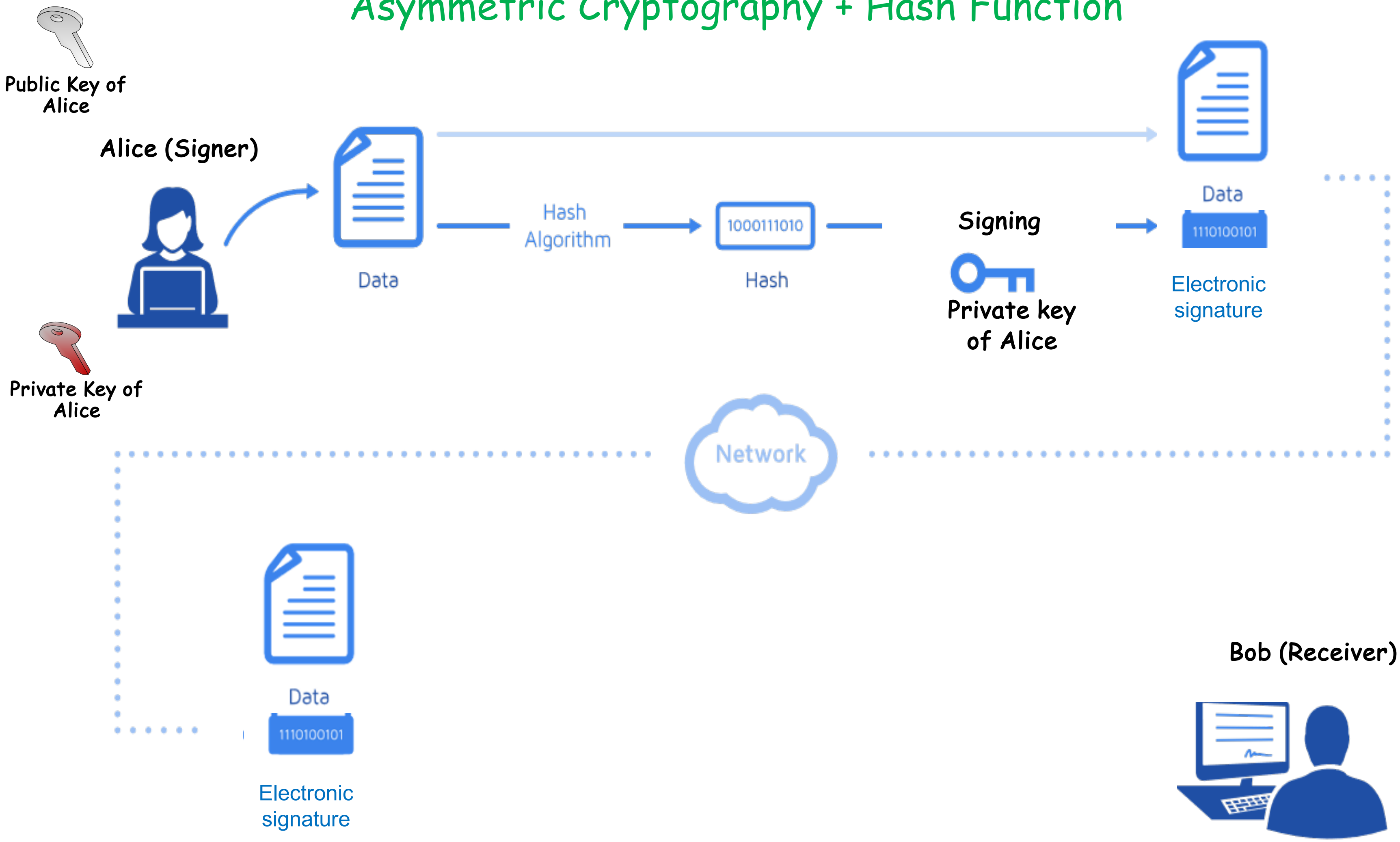


Private Key of Alice

Bob (Receiver)

Electronic Signature

Asymmetric Cryptography + Hash Function



Public Key of Alice

Alice (Signer)

Private Key of Alice

Data

Hash Algorithm

1000111010

Hash

Signing

Private key of Alice

Data

1110100101

Electronic signature

Network

Data

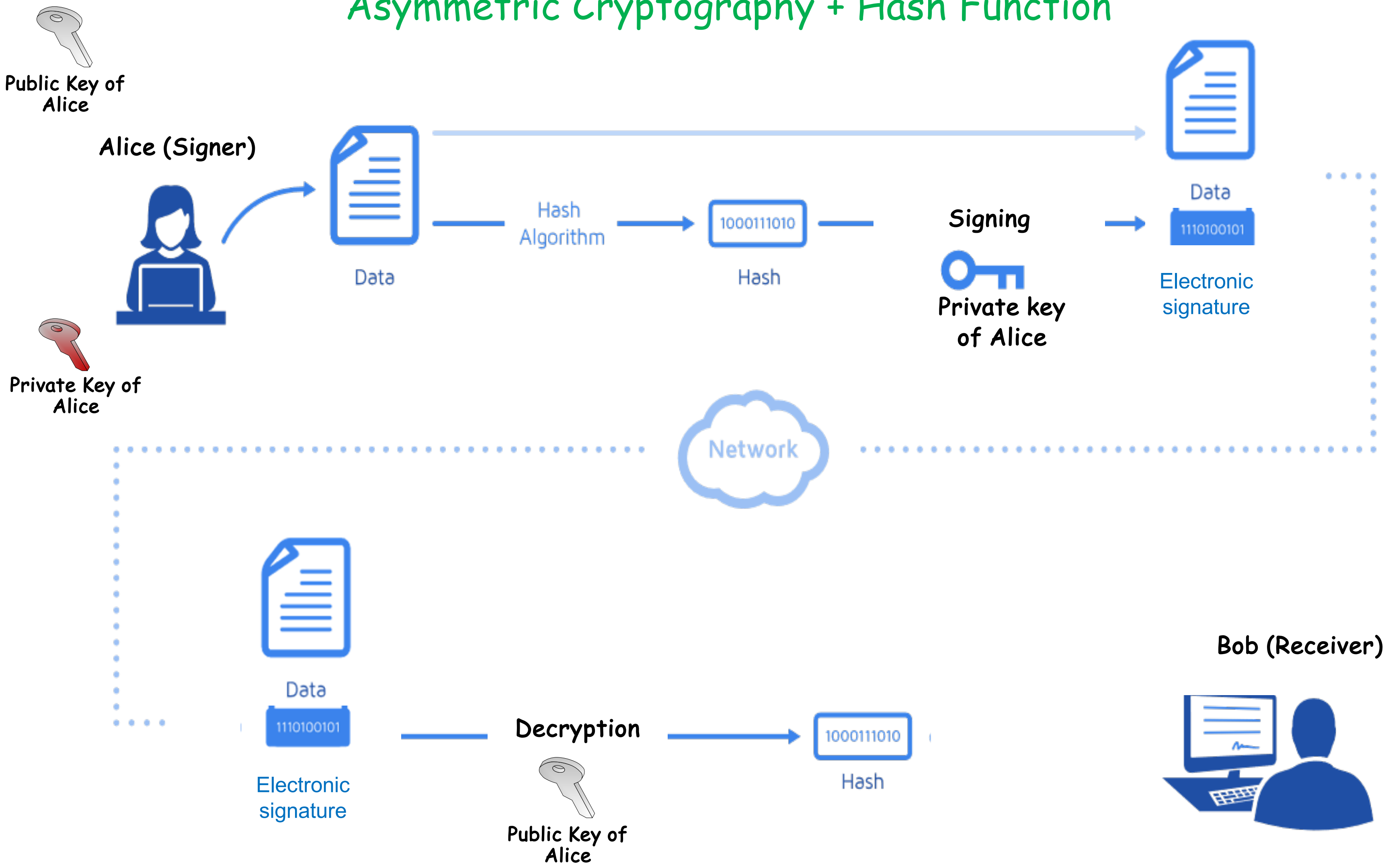
1110100101

Electronic signature

Bob (Receiver)

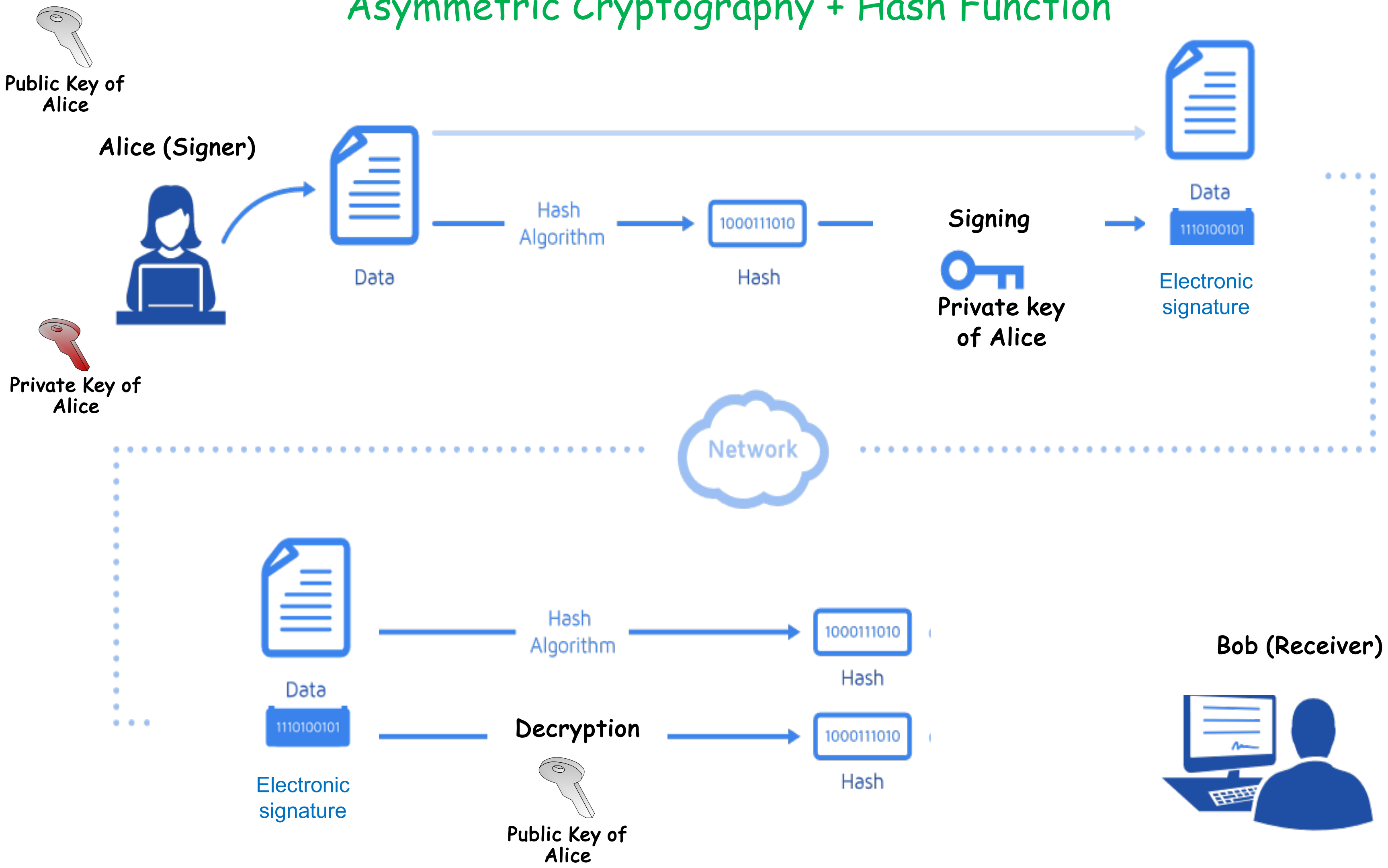
Electronic Signature

Asymmetric Cryptography + Hash Function



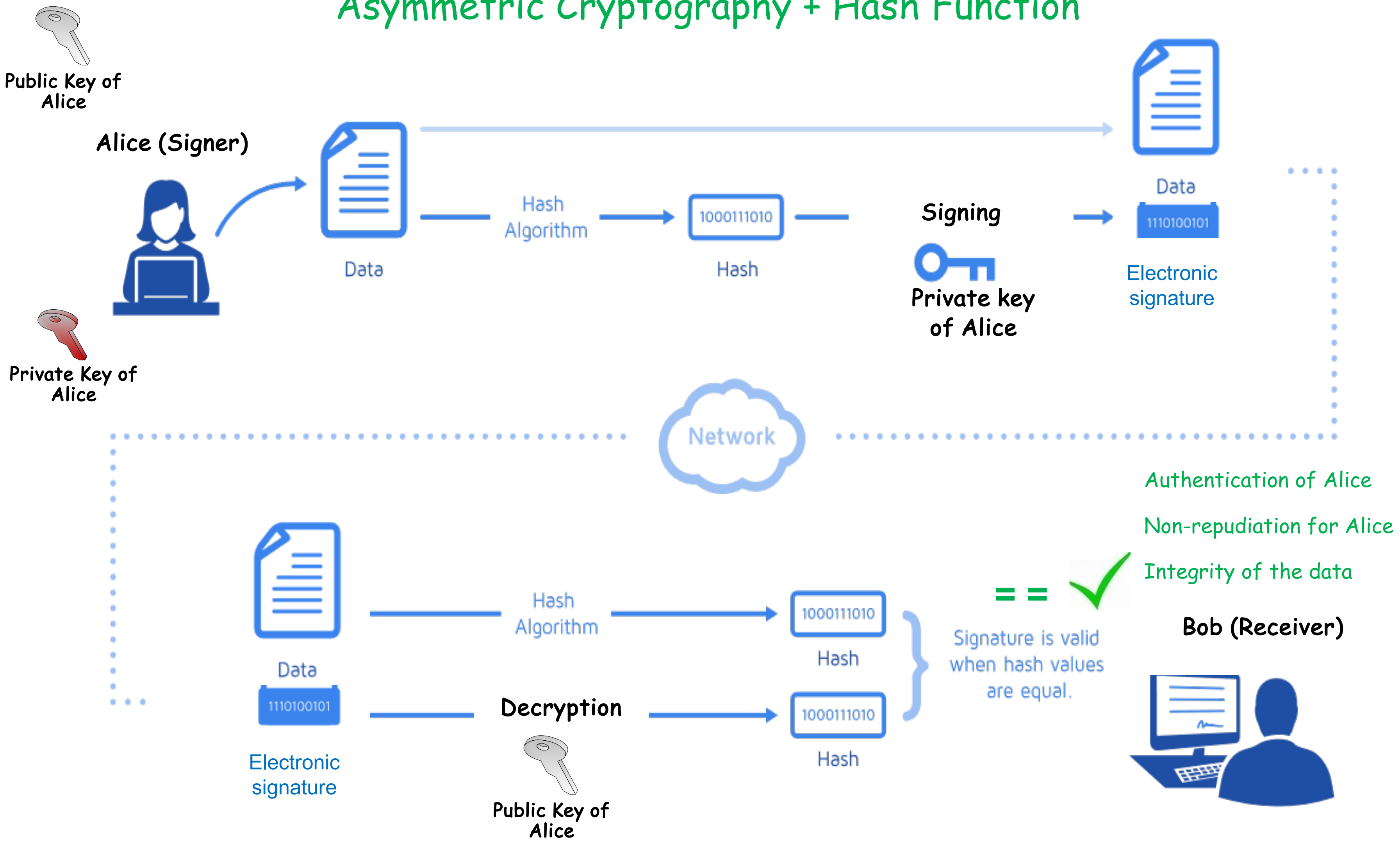
Electronic Signature

Asymmetric Cryptography + Hash Function



Electronic Signature

Asymmetric Cryptography + Hash Function



Public Key of Alice

Alice (Signer)

Private Key of Alice

Data

Hash Algorithm

1000111010

Hash

Signing

Private key of Alice

Data

1110100101

Electronic signature

Network

Authentication of Alice

Non-repudiation for Alice

Integrity of the data

Bob (Receiver)

Data

1110100101

Electronic signature

Hash Algorithm

1000111010

Hash

Decryption

Public Key of Alice

1000111010

Hash

= =

Signature is valid when hash values are equal.

25/09/2023

Electronic Signature

Asymmetric Cryptography + Hash Function

Authentication of Alice - Non-repudiation for Alice - Integrity of the data

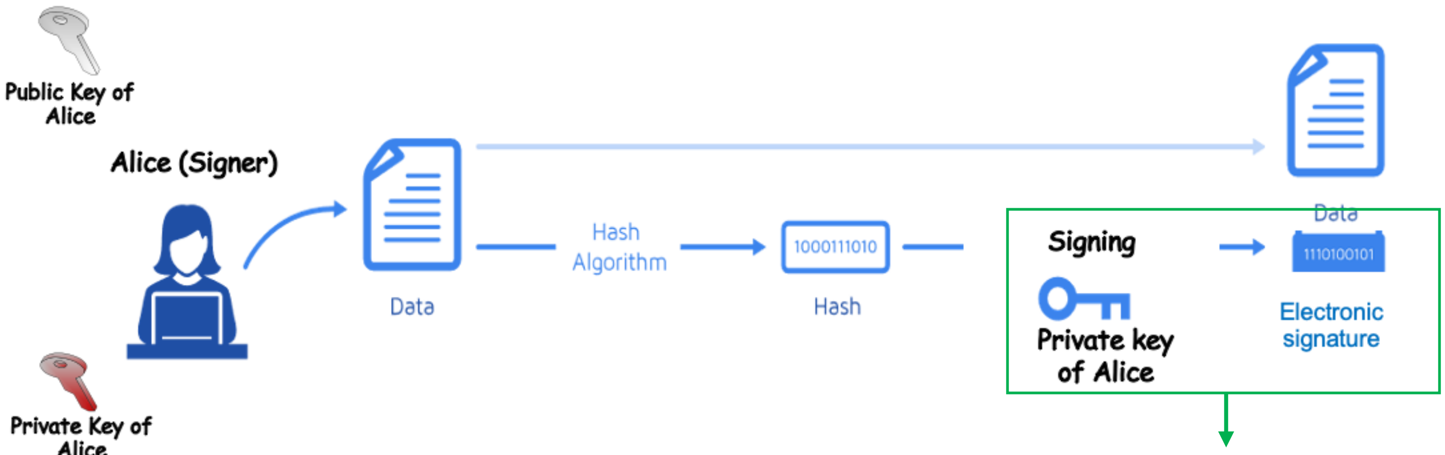
How we can explain this conclusion ?

Electronic Signature

Asymmetric Cryptography + Hash Function

Clarification:

Authentication of Alice - Non-repudiation for Alice - Integrity of the data



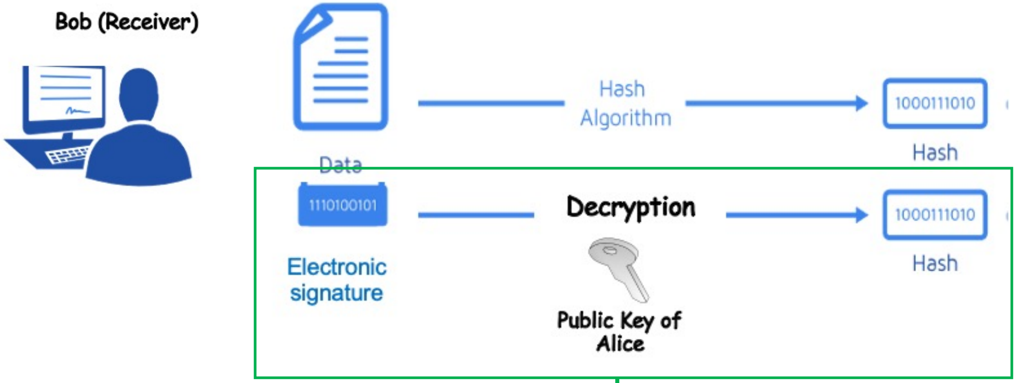
There is only Alice who can produce this signature thanks to her private key

Electronic Signature

Asymmetric Cryptography + Hash Function

Clarification:

Authentication of Alice - Non-repudiation for Alice - Integrity of the data



↓

Alice's public key matches Alice's private key

↓

✓ Authentication of Alice

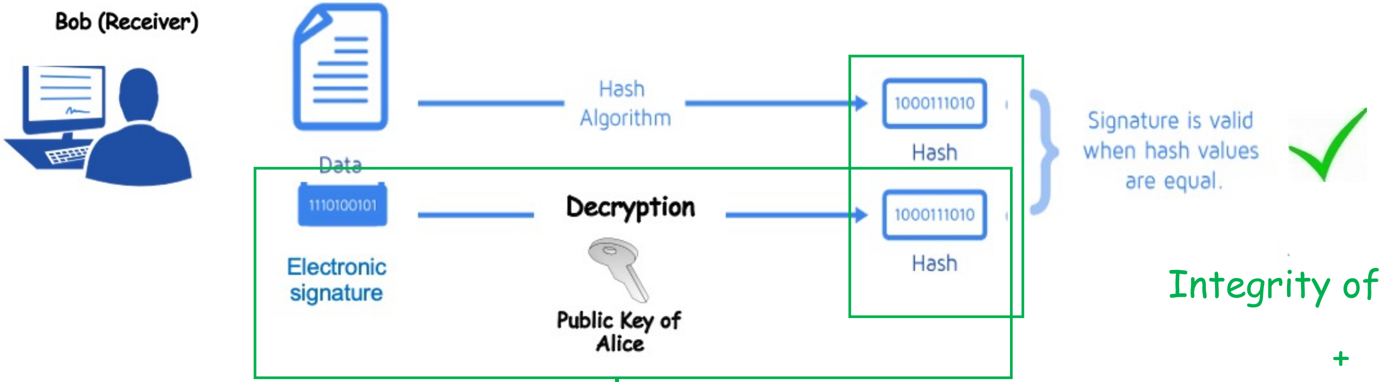
Non-repudiation for Alice

Electronic Signature

Asymmetric Cryptography + Hash Function

Clarification:

Authentication of Alice - Non-repudiation for Alice - Integrity of the data



Integrity of the data

+

Authentication of Alice

Non-repudiation for Alice

Alice's public key matches Alice's private key

✓ Authentication of Alice

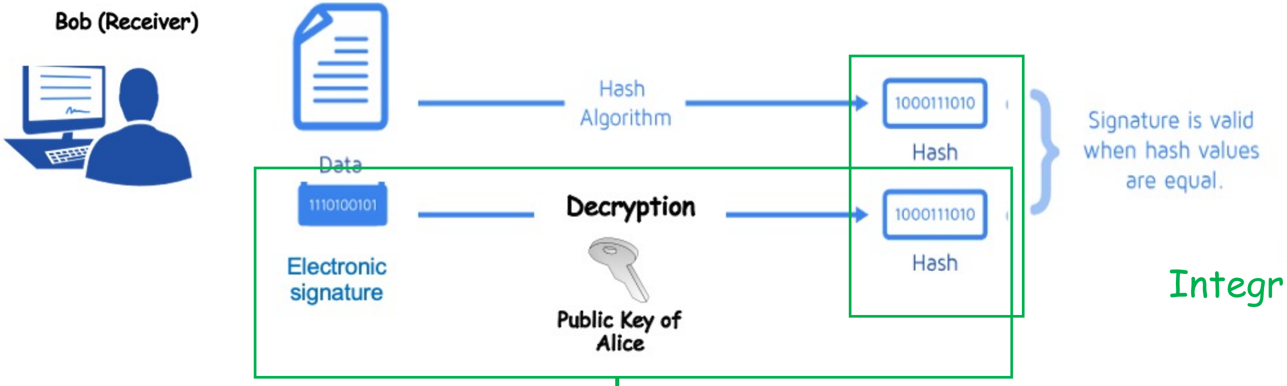
Non-repudiation for Alice

Electronic Signature

Asymmetric Cryptography + Hash Function

Clarification:

Authentication of Alice - Non-repudiation for Alice - Integrity of the data



✗

Integrity of the data ✗

+

Authentication of Alice ✗

Non-repudiation for Alice ✗

↓

Alice's public key matches Alice's private key

↓

✓ Authentication of Alice

Non-repudiation for Alice

Electronic Signature

Asymmetric Cryptography + Hash Function

Authentication of Alice - Non-repudiation for Alice - Integrity of the data

Why are the three properties insured at the same time ?

Electronic Signature

Asymmetric Cryptography + Hash Function

Why are the three properties insured at the same time?

Example: a handwritten signature on a document by Thomas

Electronic Signature

Asymmetric Cryptography + Hash Function

Example: a handwritten signature on a document by Thomas

Verification procedure

- Authentication of Thomas
- Non-repudiation for Thomas
- Integrity of the Document

Electronic Signature

Asymmetric Cryptography + Hash Function

Example: a handwritten signature on a document by Thomas

Verification procedure

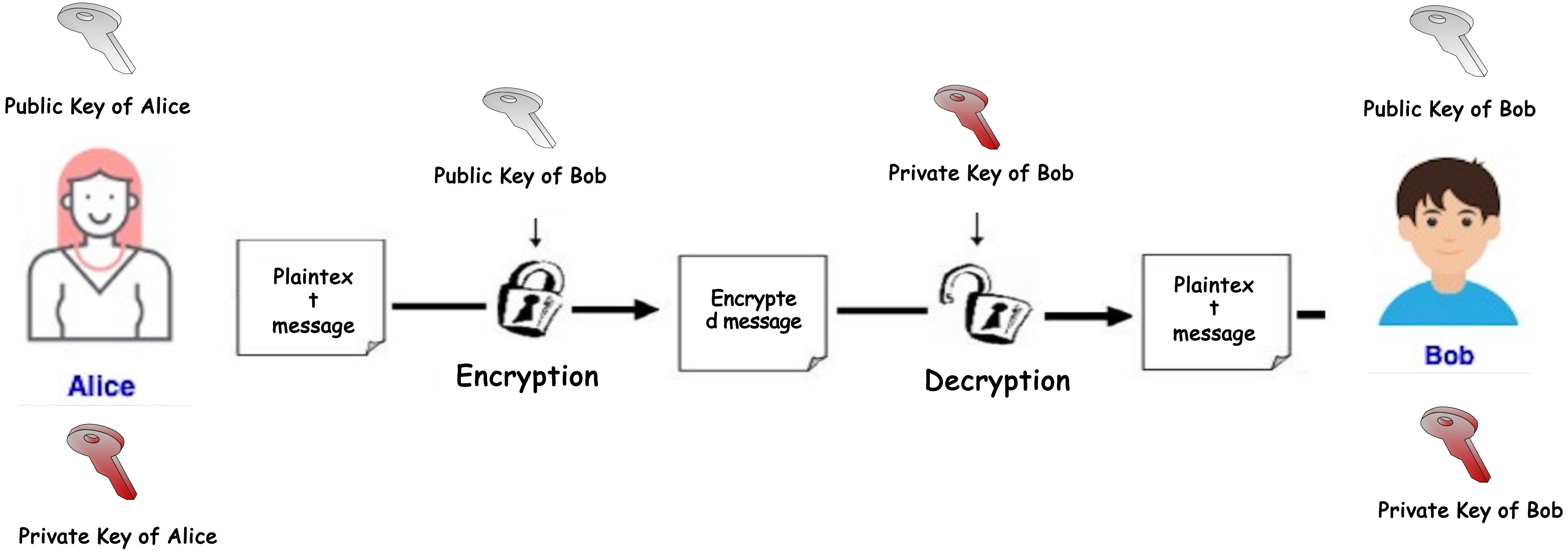
- Authentication of Thomas
- Non-repudiation for Thomas
- Integrity of the Document

The same as for an electronic signature

Electronic Certificate

Man in the Middle Attack (MITM)

How to guarantee the identity of Bob linked to his public key?



Public Key of Alice

Alice

Private Key of Alice

Plaintex
t
message

Public Key of Bob

Encryption

Encrypte
d message

Private Key of Bob

Decryption

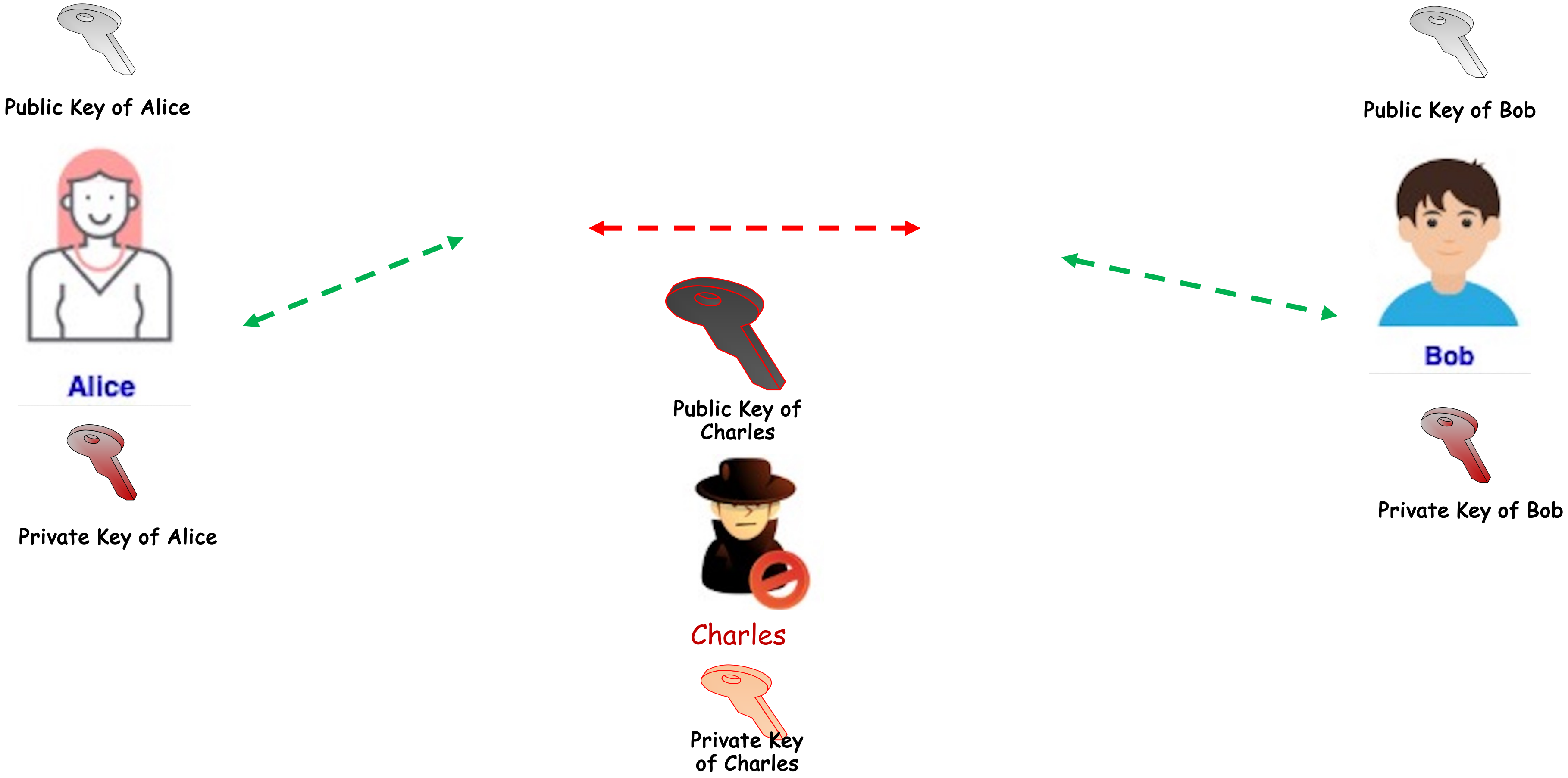
Plaintex
t
message

Public Key of Bob

Bob

Private Key of Bob

Man in the Middle Attack (MITM)



Public Key of Alice

Alice

Private Key of Alice

Public Key of Charles

Charles

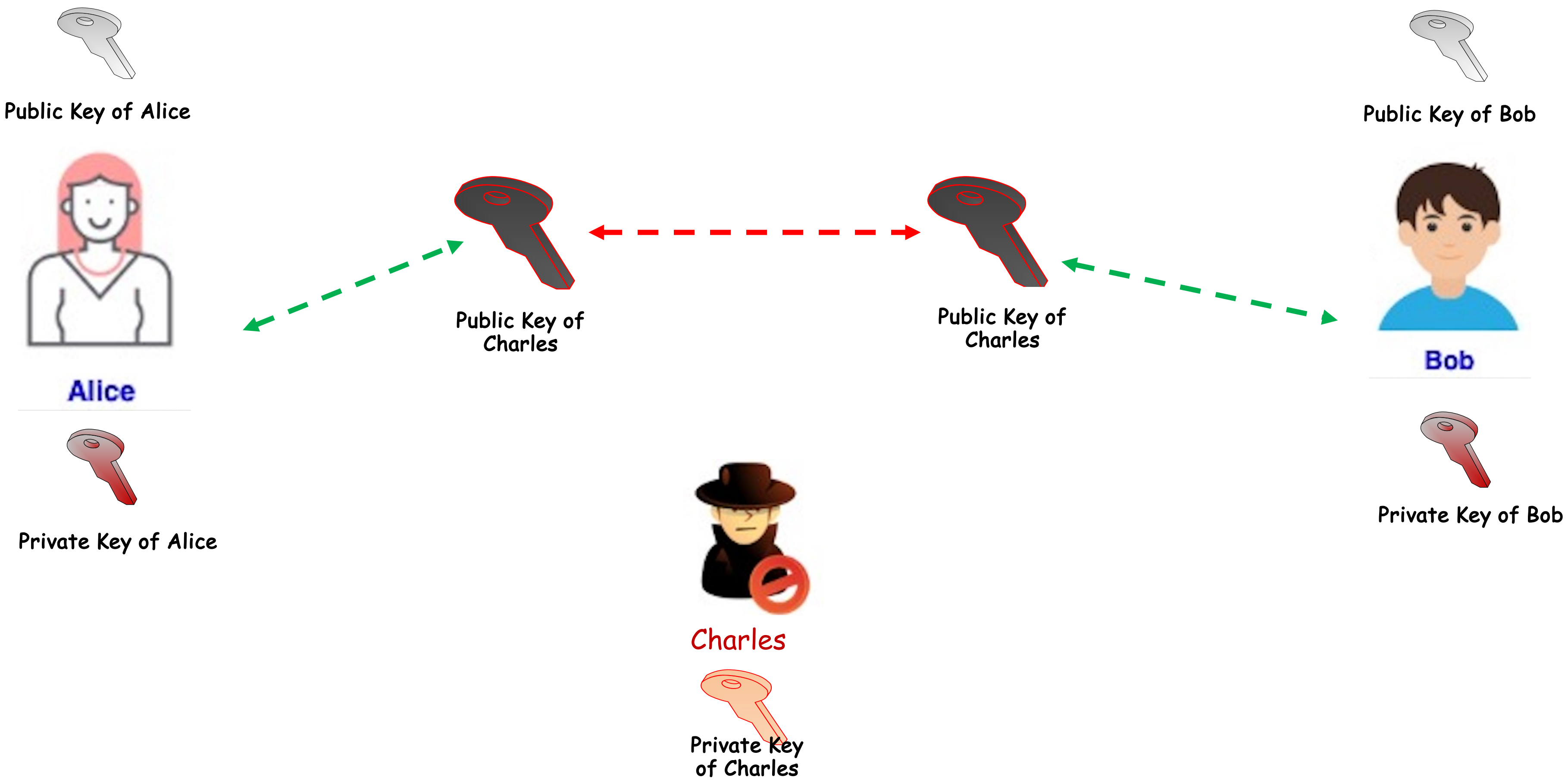
Private Key of Charles

Public Key of Bob

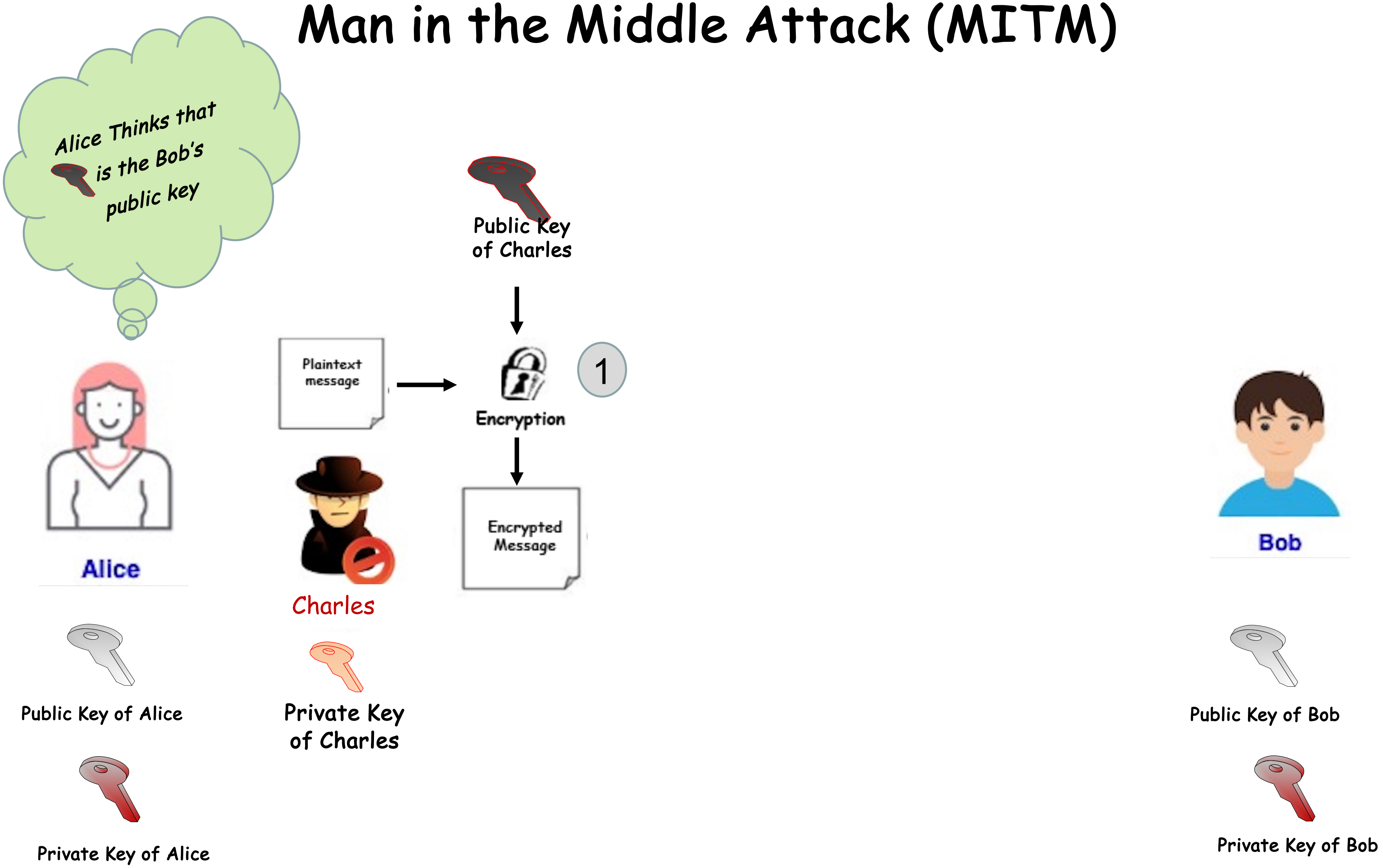
Bob

Private Key of Bob

Man in the Middle Attack (MITM)



Man in the Middle Attack (MITM)



Alice Thinks that is the Bob's public key

Public Key of Charles

Plaintext message

Encryption

1

Encrypted Message

Alice

Charles

Bob

Public Key of Alice

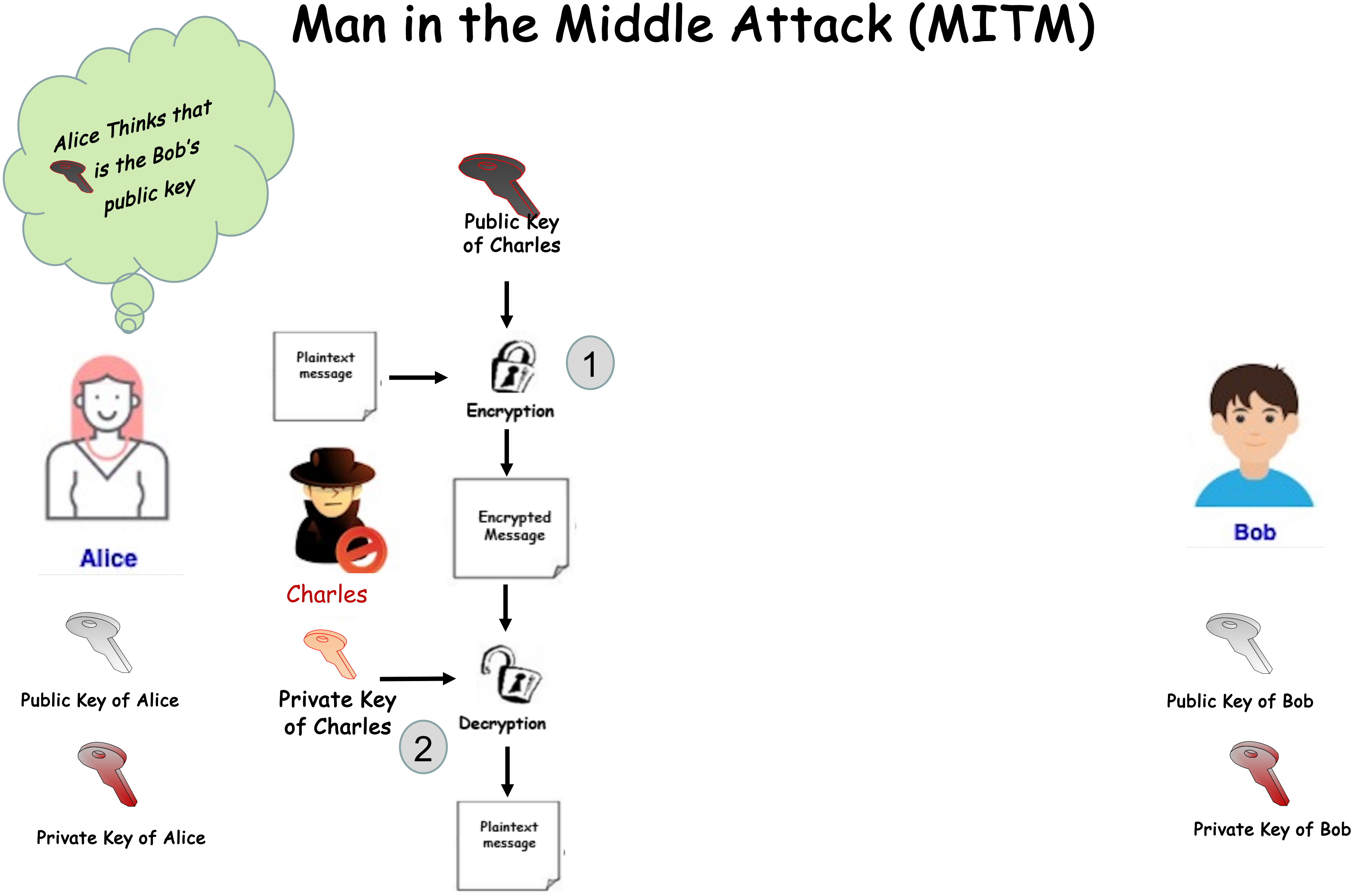
Private Key of Alice

Private Key of Charles

Public Key of Bob

Private Key of Bob

Man in the Middle Attack (MITM)



Alice Thinks that is the Bob's public key

Public Key of Charles

Plaintext message

Encryption

1

Encrypted Message

Private Key of Charles

2

Decryption

Plaintext message

Alice

Charles

Bob

Public Key of Alice

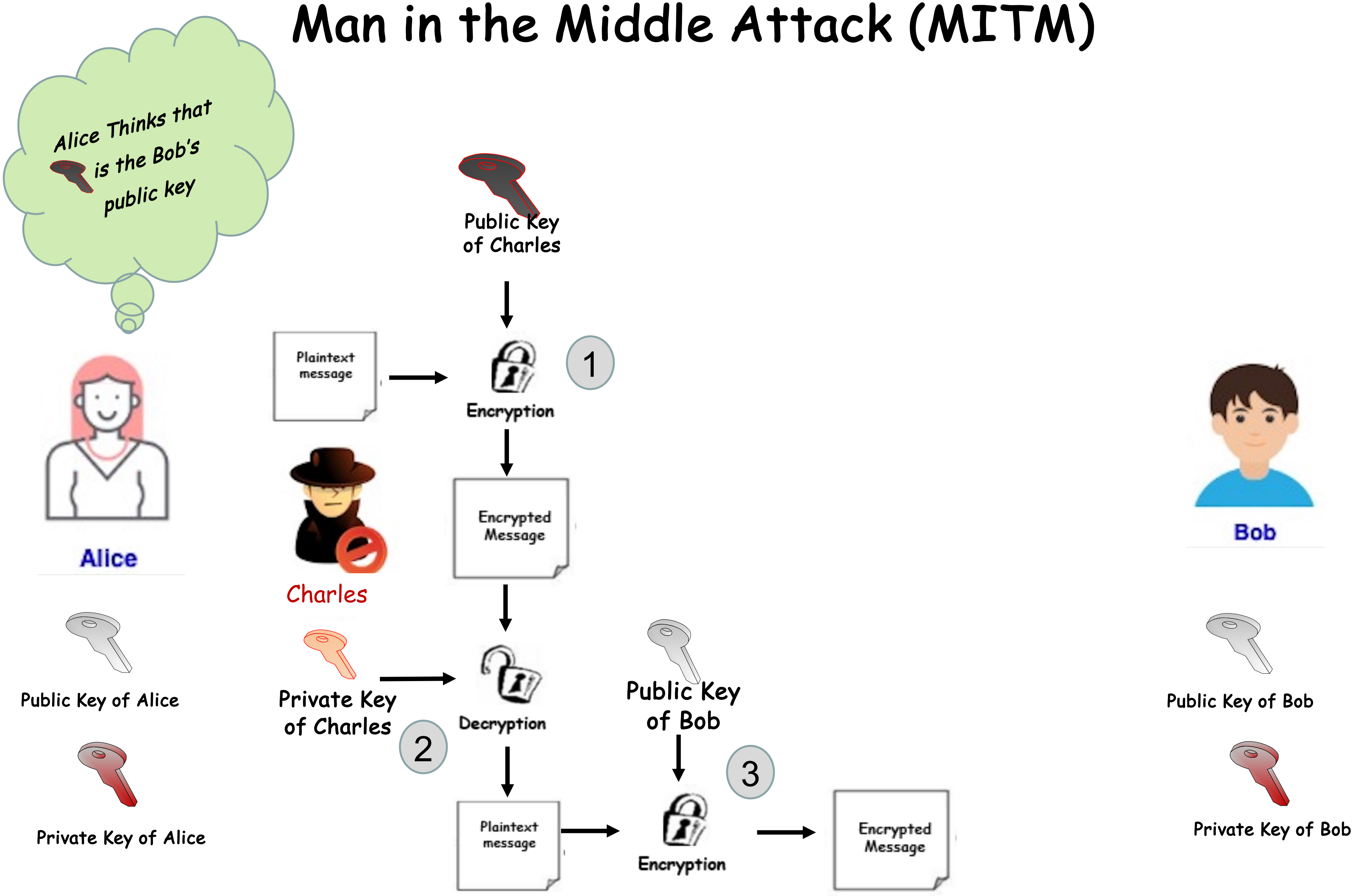
Private Key of Alice

Public Key of Bob

Private Key of Bob

25/09/2023

Man in the Middle Attack (MITM)



Alice Thinks that is the Bob's public key

Alice

Public Key of Alice

Private Key of Alice

Charles

Public Key of Charles

Plaintext message

Encryption

1

Encrypted Message

Private Key of Charles

Decryption

2

Plaintext message

Public Key of Bob

Encryption

3

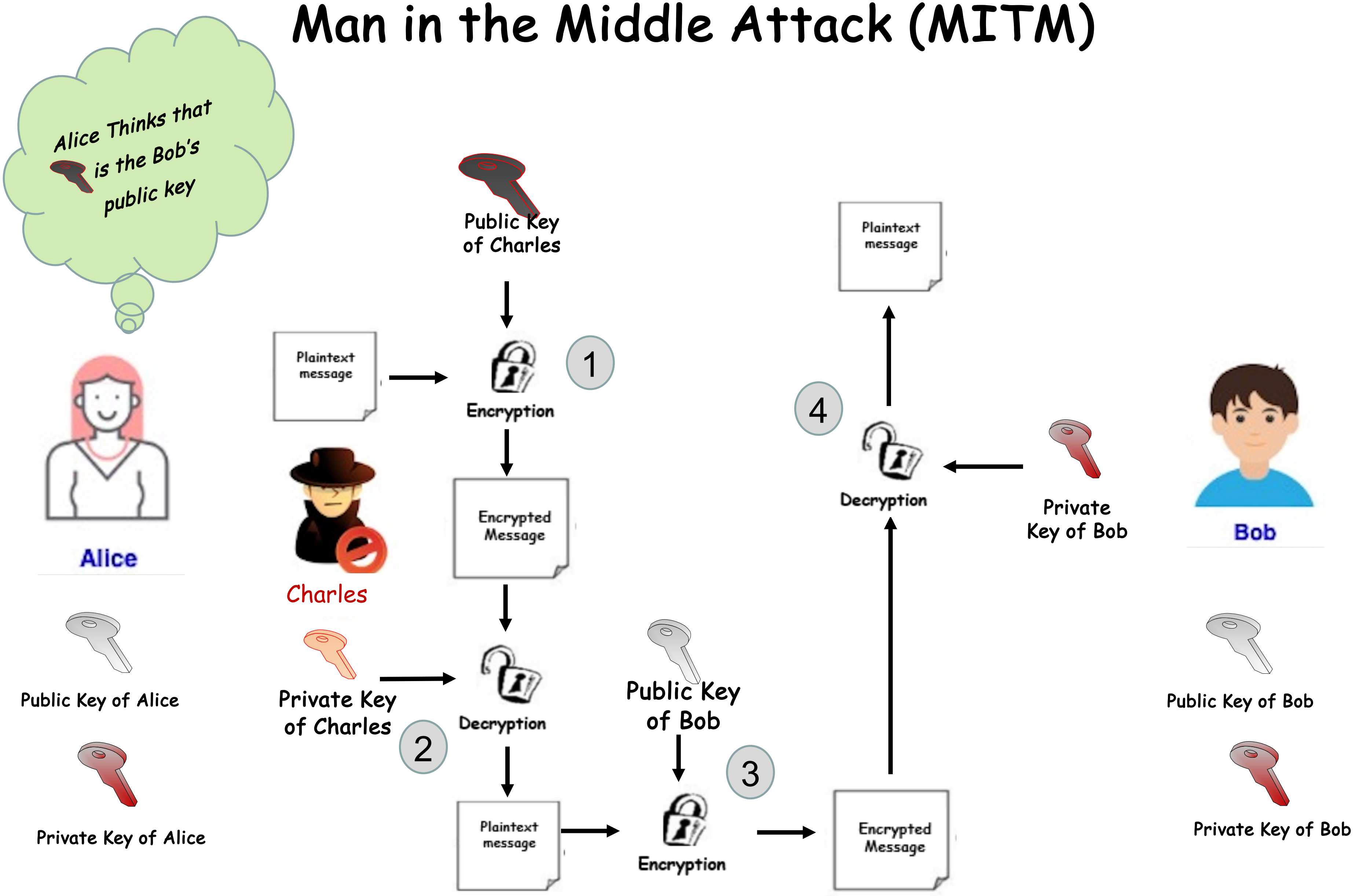
Encrypted Message

Bob

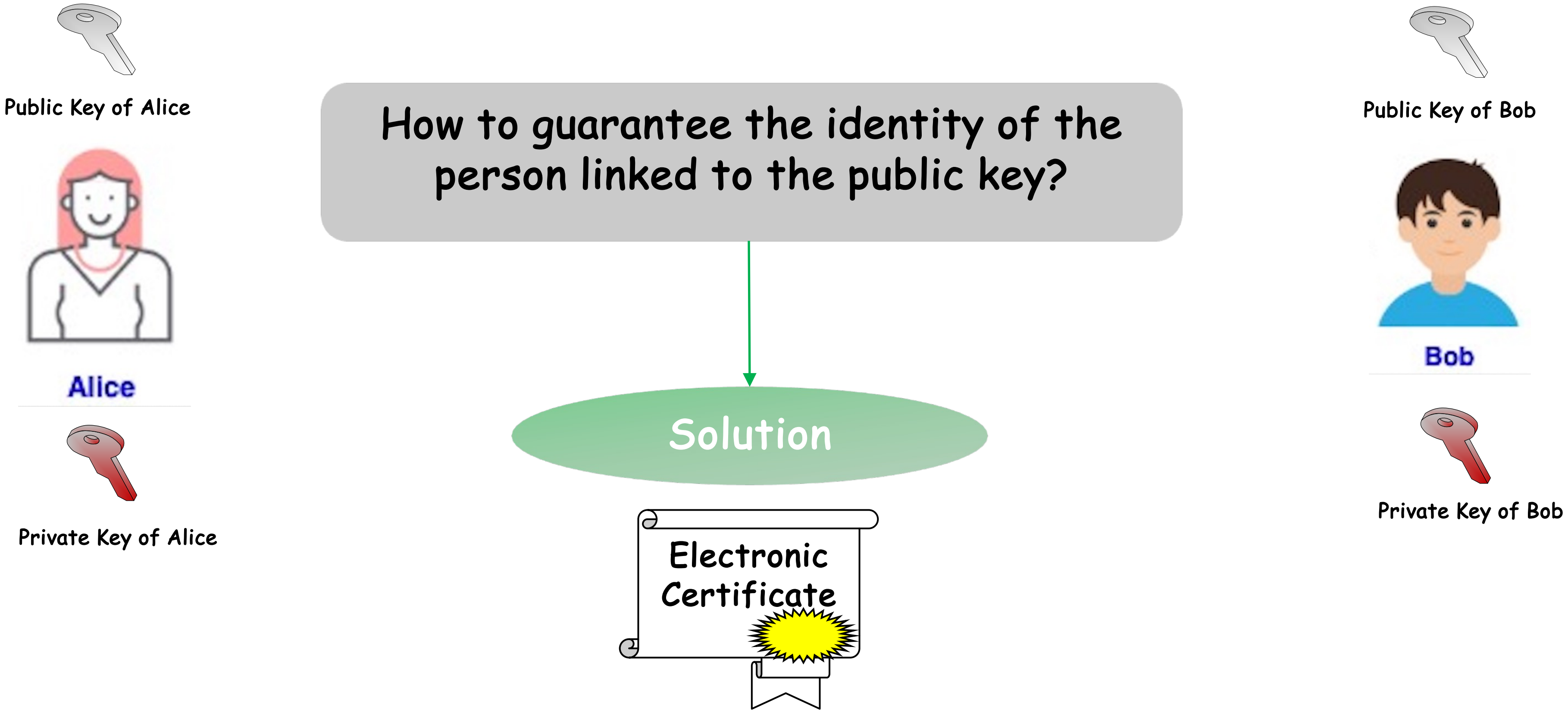
Public Key of Bob

Private Key of Bob

Man in the Middle Attack (MITM)



Man in the Middle Attack (MITM)



Public Key of Alice

Alice

Private Key of Alice

How to guarantee the identity of the person linked to the public key?

Solution

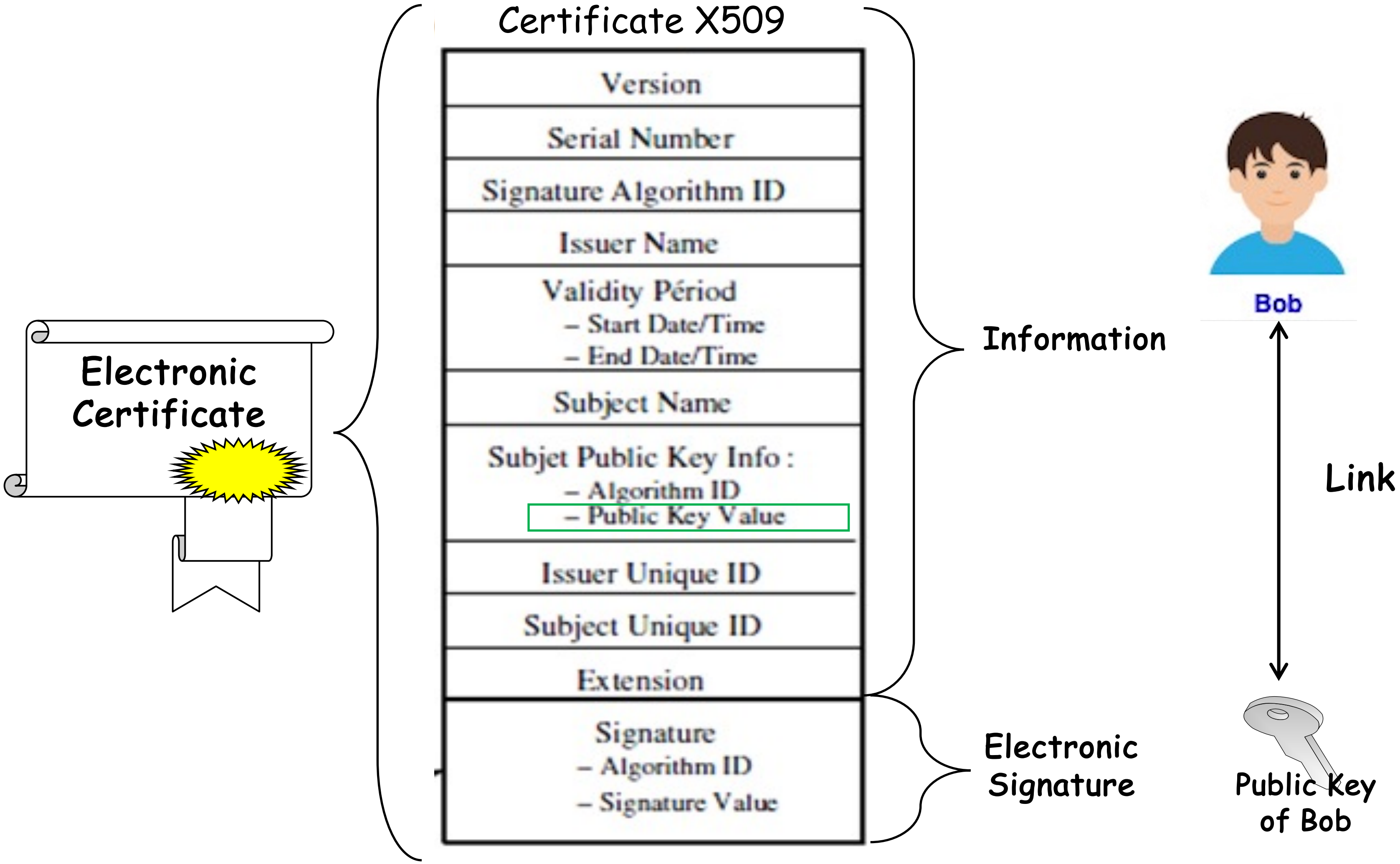
Electronic Certificate

Public Key of Bob

Bob

Private Key of Bob

Electronic Certificate



Certificate X509

Version

Serial Number

Signature Algorithm ID

Issuer Name

Validity Périod
- Start Date/Time
- End Date/Time

Subject Name

Subjet Public Key Info :
- Algorithm ID
- Public Key Value

Issuer Unique ID

Subject Unique ID

Extension

Signature
- Algorithm ID
- Signature Value

pk: public key
sk: (private) secret key
Cert: certificate

Electronic Certificate

Certification Authority (CA)

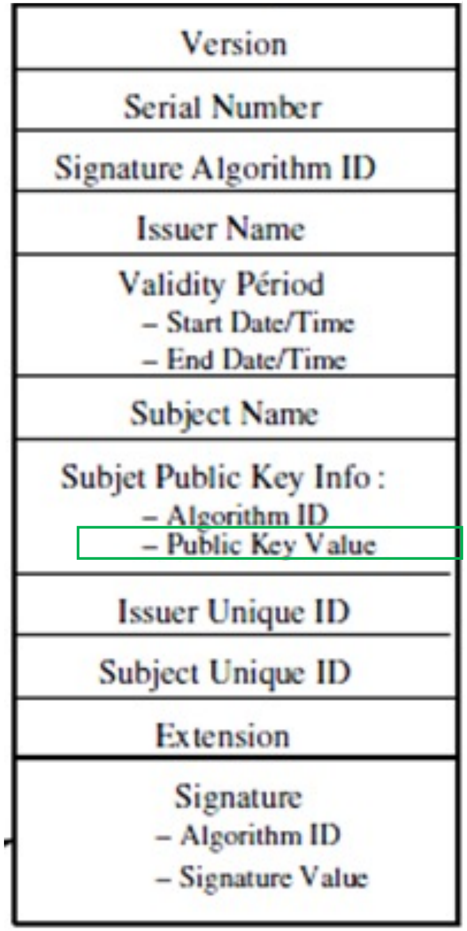
- CA has the role of a Trusted Third Party

- It has a key pair:
    - pk(CA)/sk(CA): self-generated

- Everyone should trust CA without the need of any verification

- Alice and Bob should trust this CA

Certificate X509



Certificate X509
Version
Serial Number
Signature Algorithm ID
Issuer Name
Validity Périod – Start Date/Time – End Date/Time
Subject Name
Subjet Public Key Info : – Algorithm ID – Public Key Value
Issuer Unique ID
Subject Unique ID
Extension
Signature – Algorithm ID – Signature Value

pk: public key
sk: (private) secret key
Cert: certificate

Electronic Certificate

Certification Authority (CA)

- CA has the role of a Trusted Third Party

- It has a key pair:
    - pk(CA)/sk(CA): self-generated

- Everyone should trust CA without the need of any verification

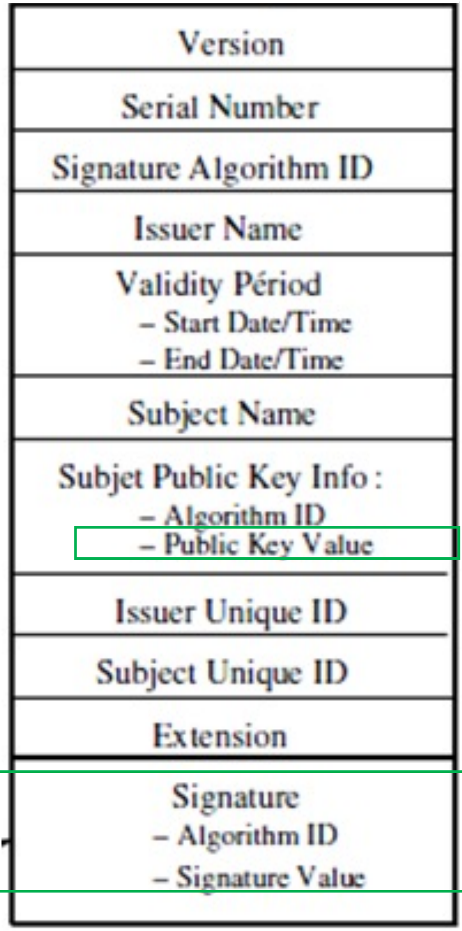
- Alice and Bob should trust this CA

- Alice and Bob have already in their Database pk(CA)

- CA will use its sk(CA) to generate the signature of a certificate

- Alice and Bob can use without any problem pk(CA)

Certificate X509

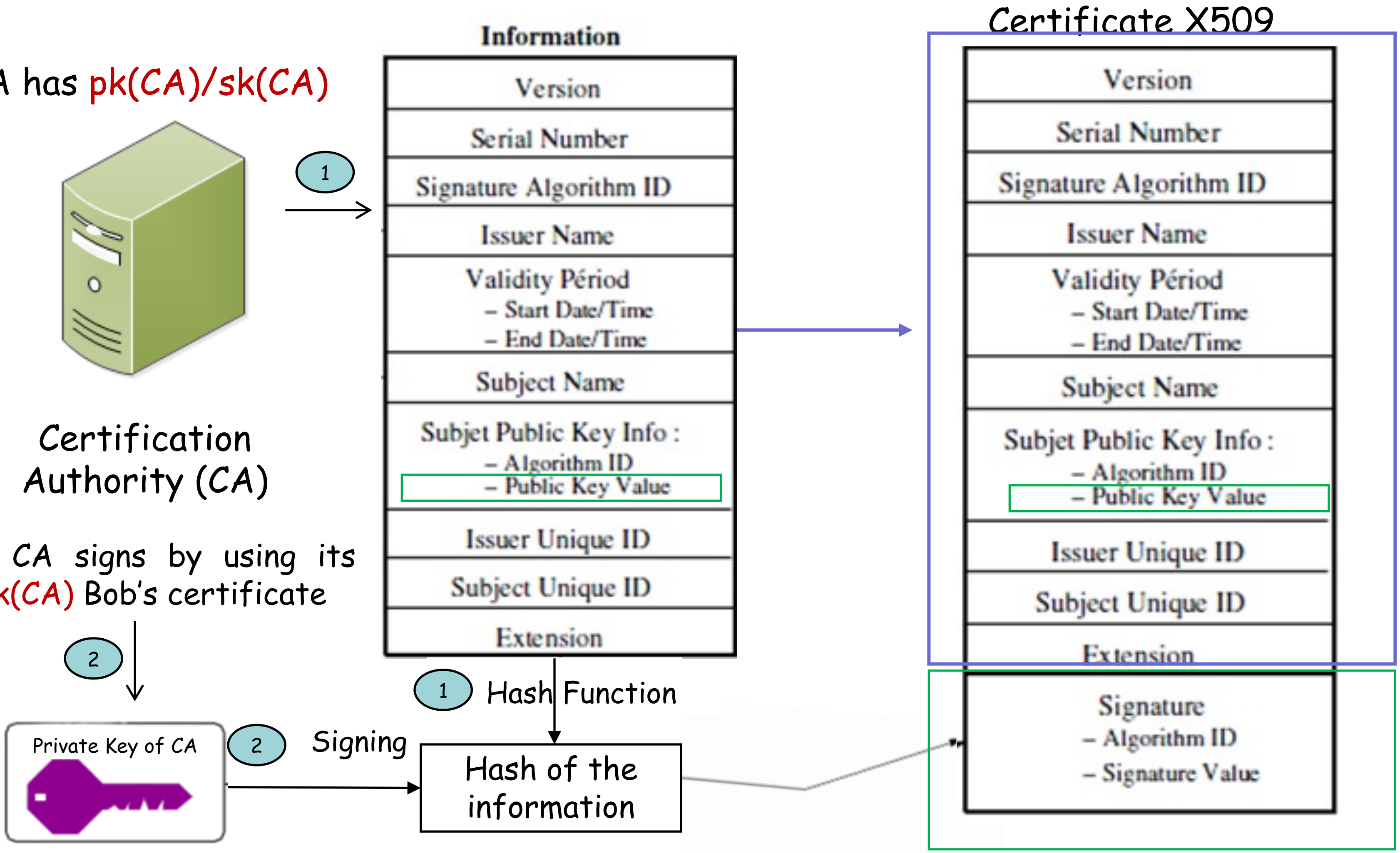


Version
Serial Number
Signature Algorithm ID
Issuer Name
Validity Périod – Start Date/Time – End Date/Time
Subject Name
Subjet Public Key Info : – Algorithm ID – Public Key Value
Issuer Unique ID
Subject Unique ID
Extension
Signature – Algorithm ID – Signature Value

pk: public key
sk: (private) secret key

Electronic Certificate

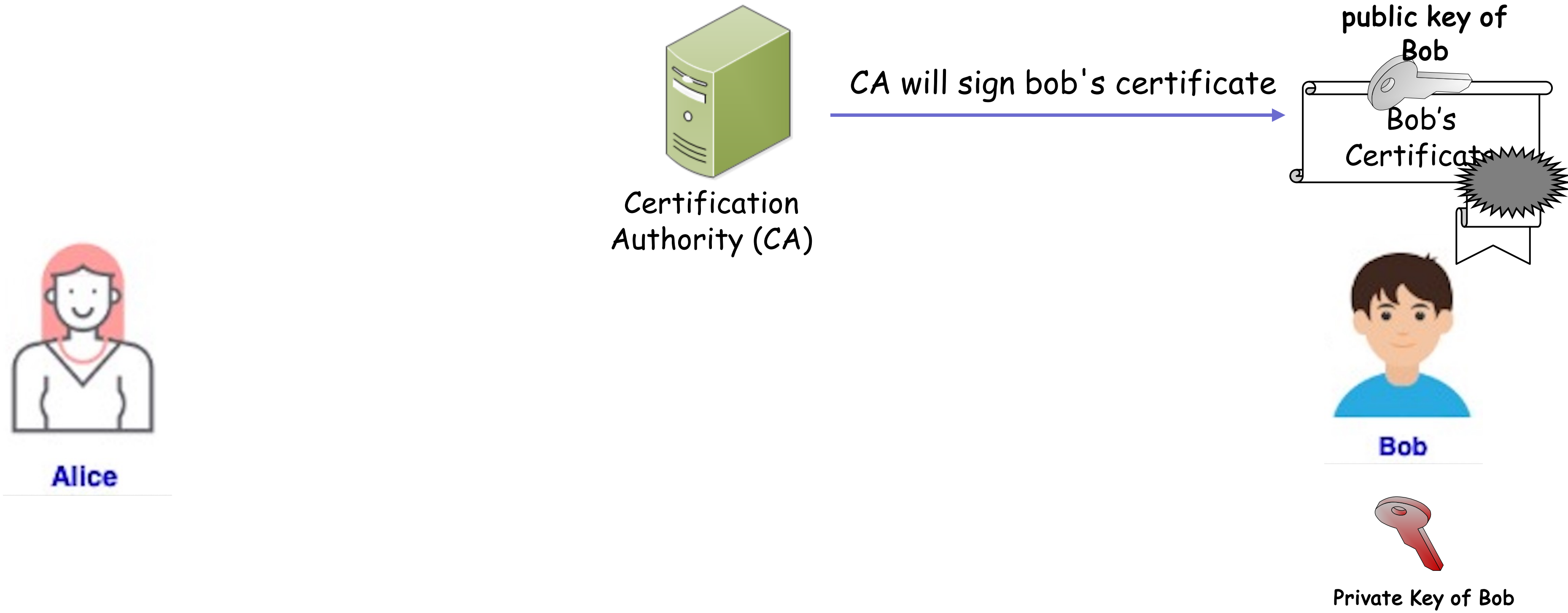
-CA has pk(CA)/sk(CA)



Certification Authority (CA)

- CA signs by using its sk(CA) Bob's certificate

Electronic Certificate



Certification Authority (CA)

CA will sign bob's certificate

public key of Bob

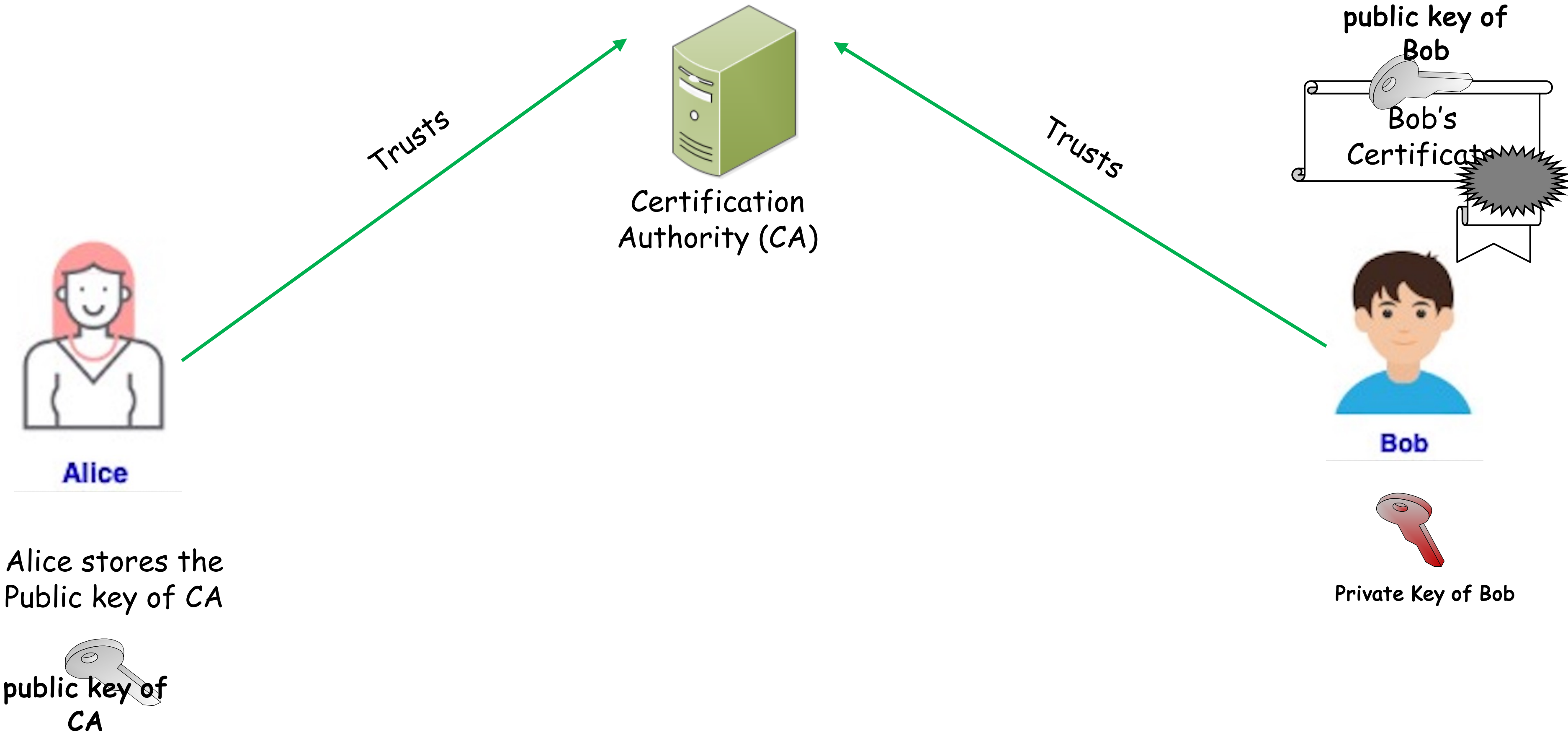
Bob's Certificate

Bob

Alice

Private Key of Bob

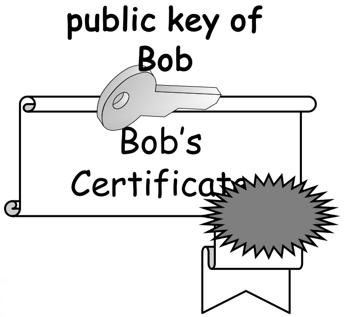
Electronic Certificate



Alice stores the Public key of CA

Electronic Certificate

How will Alice proceed to use Bob's public key after obtaining Bob's certificate?



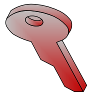
Alice stores the Public key of CA

* Before using the public key of Bob,
Alice must verify the certificate of bob

* Alice will verify the certificate of Bob to confirm that Bob is link to this certificate

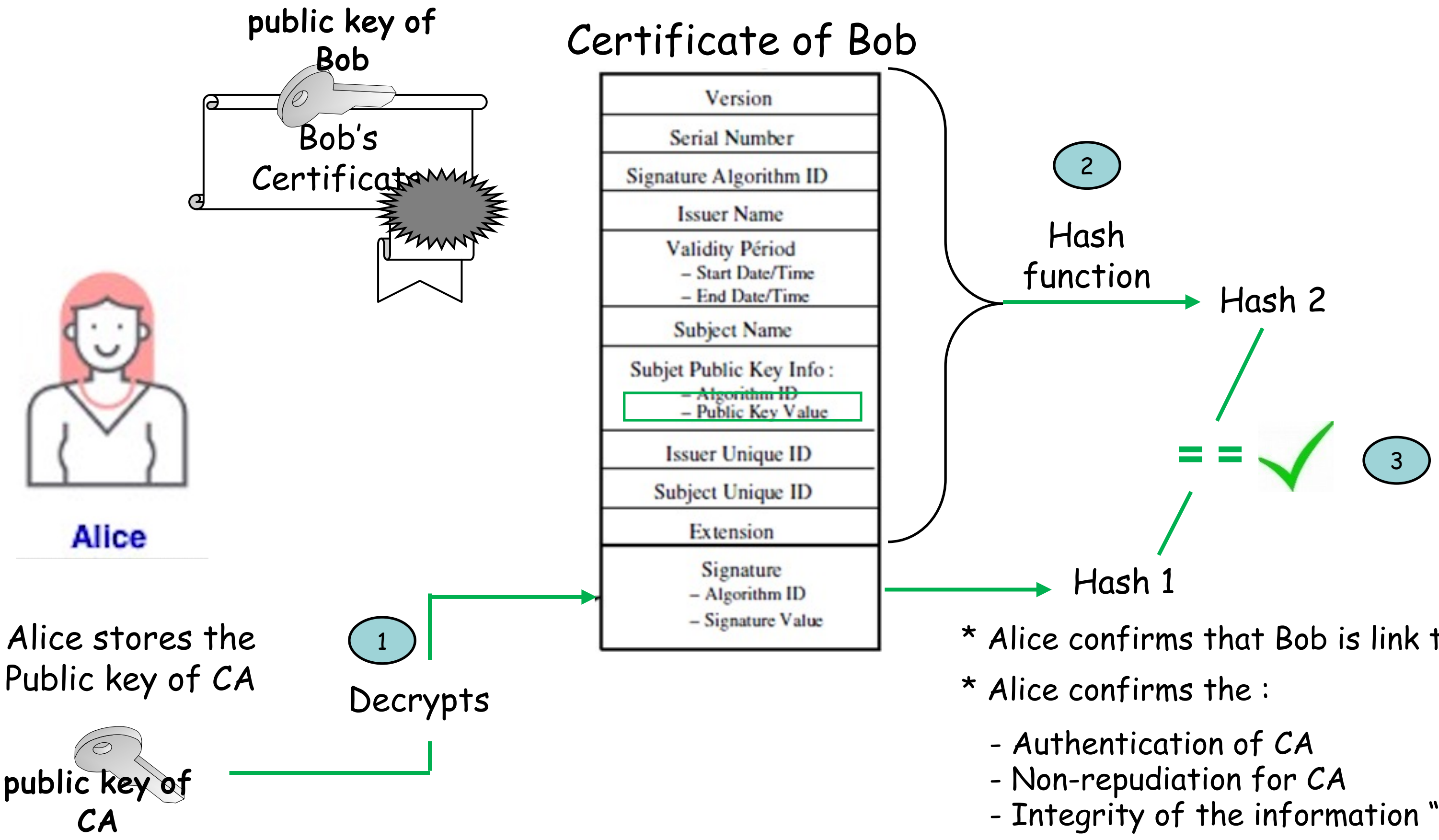
* Alice will verify the certificate of Bob to confirm the :
- Authentication of CA
- Non-repudiation for CA
- Integrity of the information “the public key of Bob”

* Alice will verify the certificate of Bob by verifying its signature thanks **to the public of CA**



Electronic Certificate

How will Alice proceed to use Bob's public key after obtaining Bob's certificate?



public key of Bob

Bob's Certificate

Certificate of Bob

- Version
- Serial Number
- Signature Algorithm ID
- Issuer Name
- Validity Période
  - – Start Date/Time
  - – End Date/Time
- Subject Name
- Subjet Public Key Info :
  - – Algorithm ID
  - – Public Key Value
- Issuer Unique ID
- Subject Unique ID
- Extension
- Signature
  - – Algorithm ID
  - – Signature Value

Alice stores the Public key of CA

public key of CA

* Alice confirms that Bob is link to this certificate

* Alice confirms the :

- Authentication of CA
- Non-repudiation for CA
- Integrity of the information “the public key of Bob”

* Alice can then use the public key of Bob !

pk: public key
sk: (private) secret key
Cert: certificate

Electronic Certificate

More precisions about a certification authority

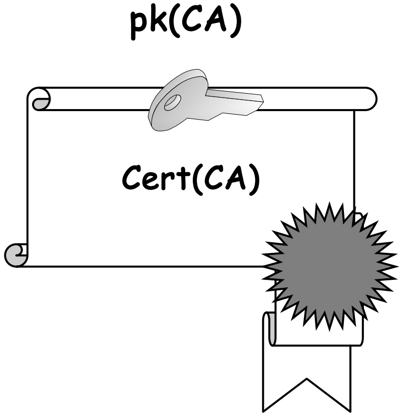
pk(CA)
Certification Authority (CA)
sk(CA)

- CA uses its sk(CA) to generate a certificate for a person/server

+

- CA uses its sk(CA) to generate a certificate for itself

--> CA has Cert(CA): **<u>self-Signed</u>**



Any person trusts CA stores Cert(CA) : but not only pk(CA)

Alice then stores Cert(CA)

Alice can use directly pk(CA) obtained from Cert(CA)

pk: public key
sk: (private) secret key
Cert: certificate

Electronic Certificate

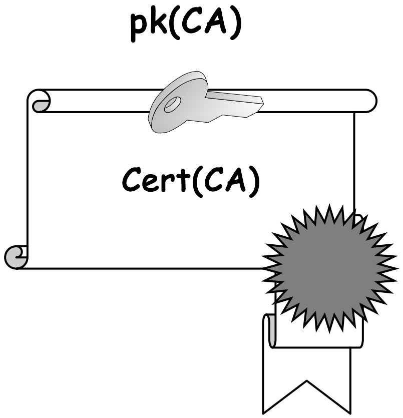
More precisions about a certification authority

- CA uses its sk(CA) to generate a certificate for a person/server

+

- CA uses its sk(CA) to generate a certificate for itself

--> CA has Cert(CA): <u>**self-Signed**</u>



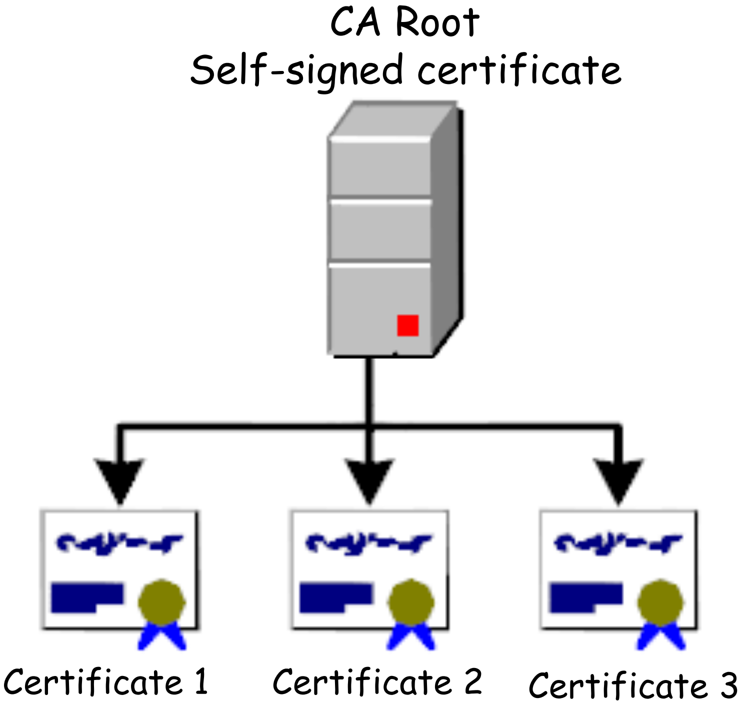
Any person trusts CA stores Cert(CA) : but not only pk(CA)

Alice then stores Cert(CA)

Alice can use directly pk(CA) obtained from Cert(CA)

Trust Models

1- Root CA Model

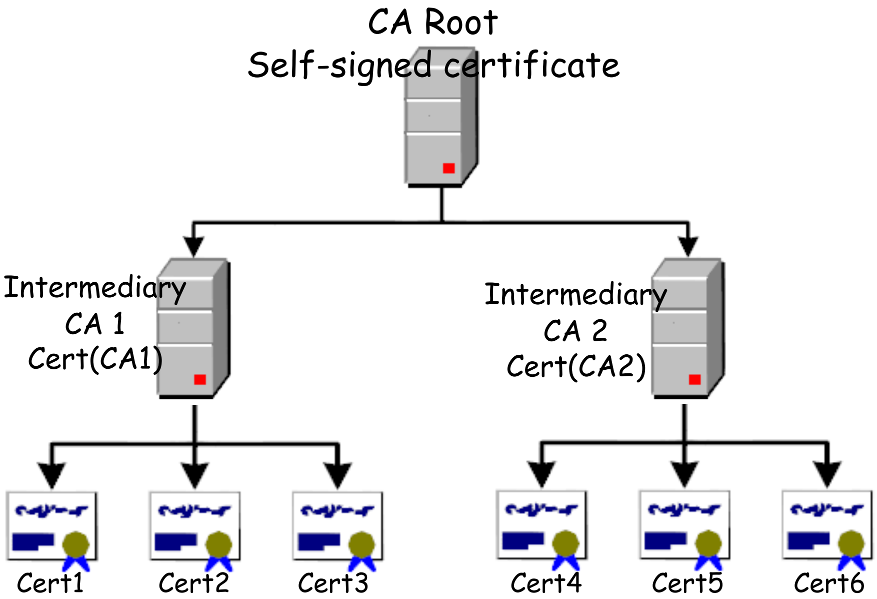


1- The CA root has its private/public keys and its self-signed certificate which contains its public key. The CA signs by using its private key the certificates 1, 2, 3

2- You need to trust the CA root and its public key. The public key of CA is used to verity the signatures of the certificates 1, 2 , 3

Trust Models

2- Hierarchical Model



A chain of Trust

1- The CA root has its private/public keys and its self-signed certificate which contains its public key.

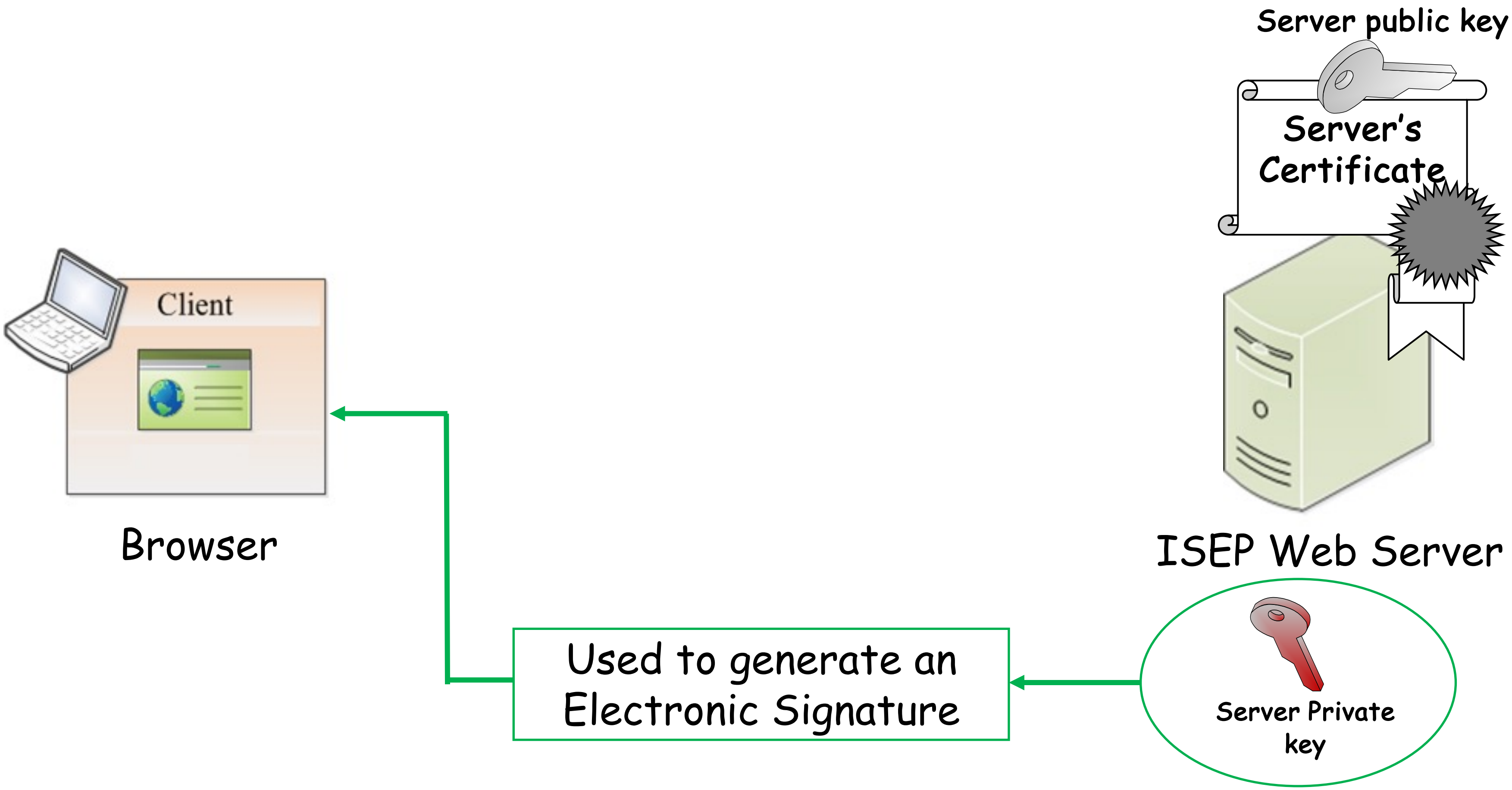
2- The CA root generates for each of CA1 and CA2 their key pair.

3- The CA root signs by using its private key the certificates of CA1 and CA2.

2- You need to trust AT LEAST the CA root and may be or not CA1 and CA2.

Presentation of a real certificate

Example: https://www.isep.fr



Server public key

Server's Certificate

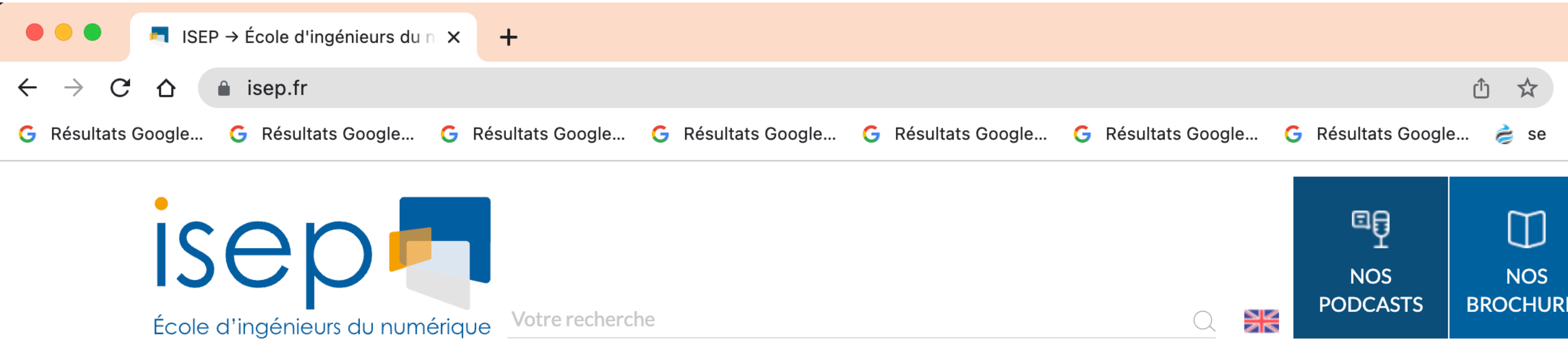
Browser

ISEP Web Server

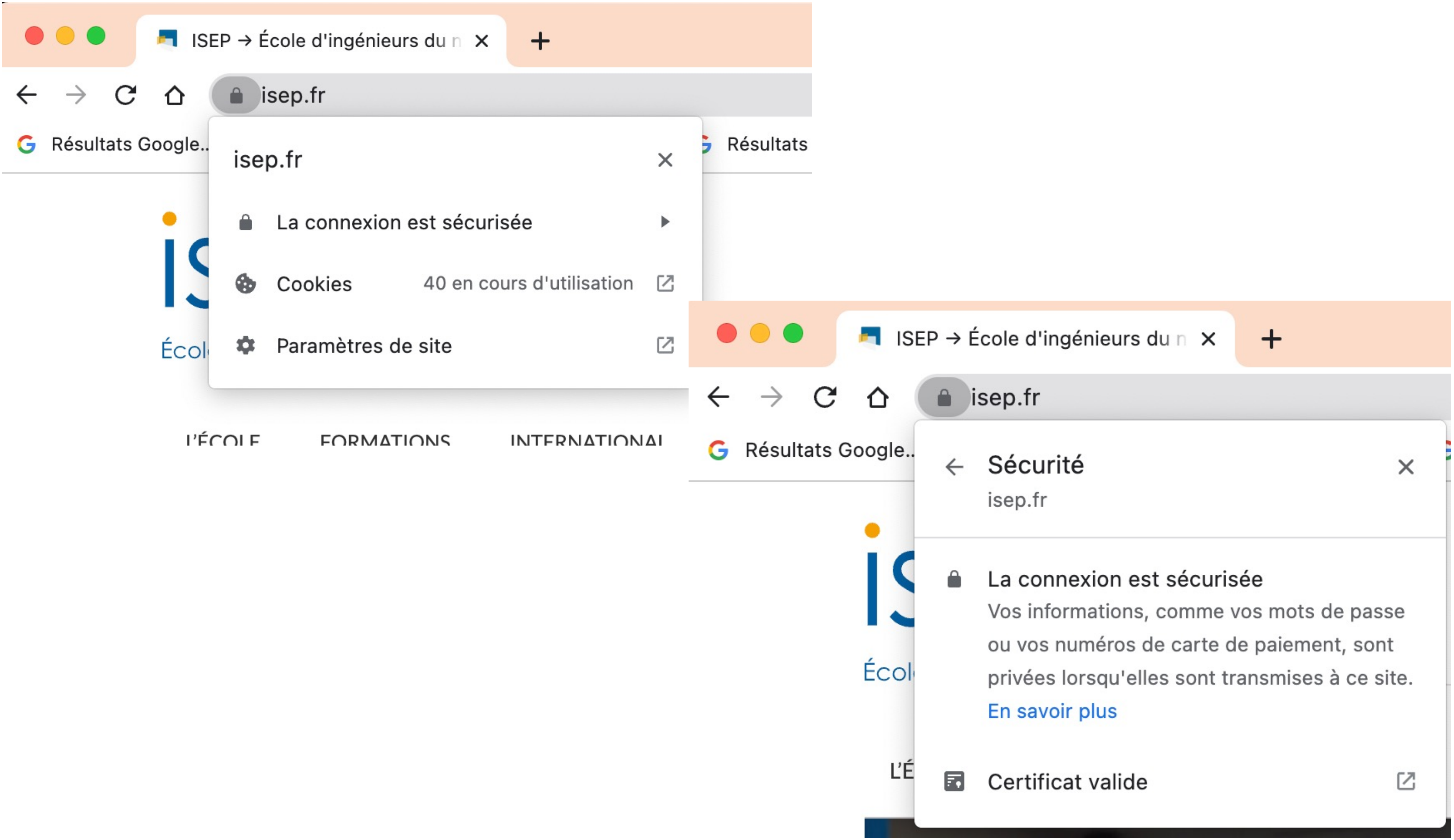
Used to generate an Electronic Signature

Server Private key

Example: https://www.isep.fr



Example: https://www.isep.fr



ISEP → École d'ingénieurs du n ✕ +

isep.fr

Résultats Google...

isep.fr ✕

La connexion est sécurisée

Cookies 40 en cours d'utilisation

Paramètres de site

ISEP → École d'ingénieurs du n ✕ +

isep.fr

Résultats Google...

Sécurité ✕

isep.fr

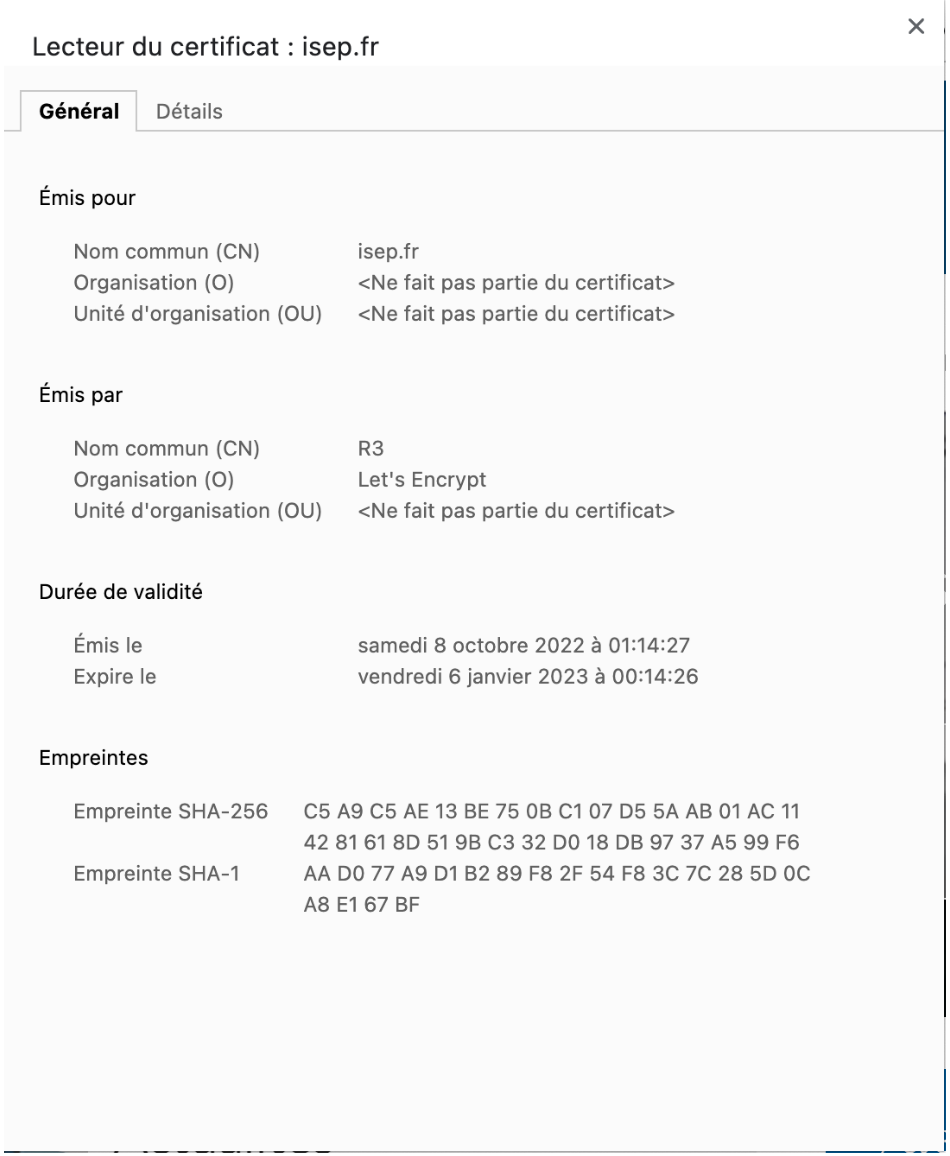
La connexion est sécurisée

Vos informations, comme vos mots de passe ou vos numéros de carte de paiement, sont privées lorsqu'elles sont transmises à ce site.

En savoir plus

Certificat valide

Example: https://www.isep.fr



Lecteur du certificat : isep.fr

Émis pour

Nom commun (CN)	isep.fr
Organisation (O)	<Ne fait pas partie du certificat>
Unité d'organisation (OU)	<Ne fait pas partie du certificat>

Émis par

Nom commun (CN)	R3
Organisation (O)	Let's Encrypt
Unité d'organisation (OU)	<Ne fait pas partie du certificat>

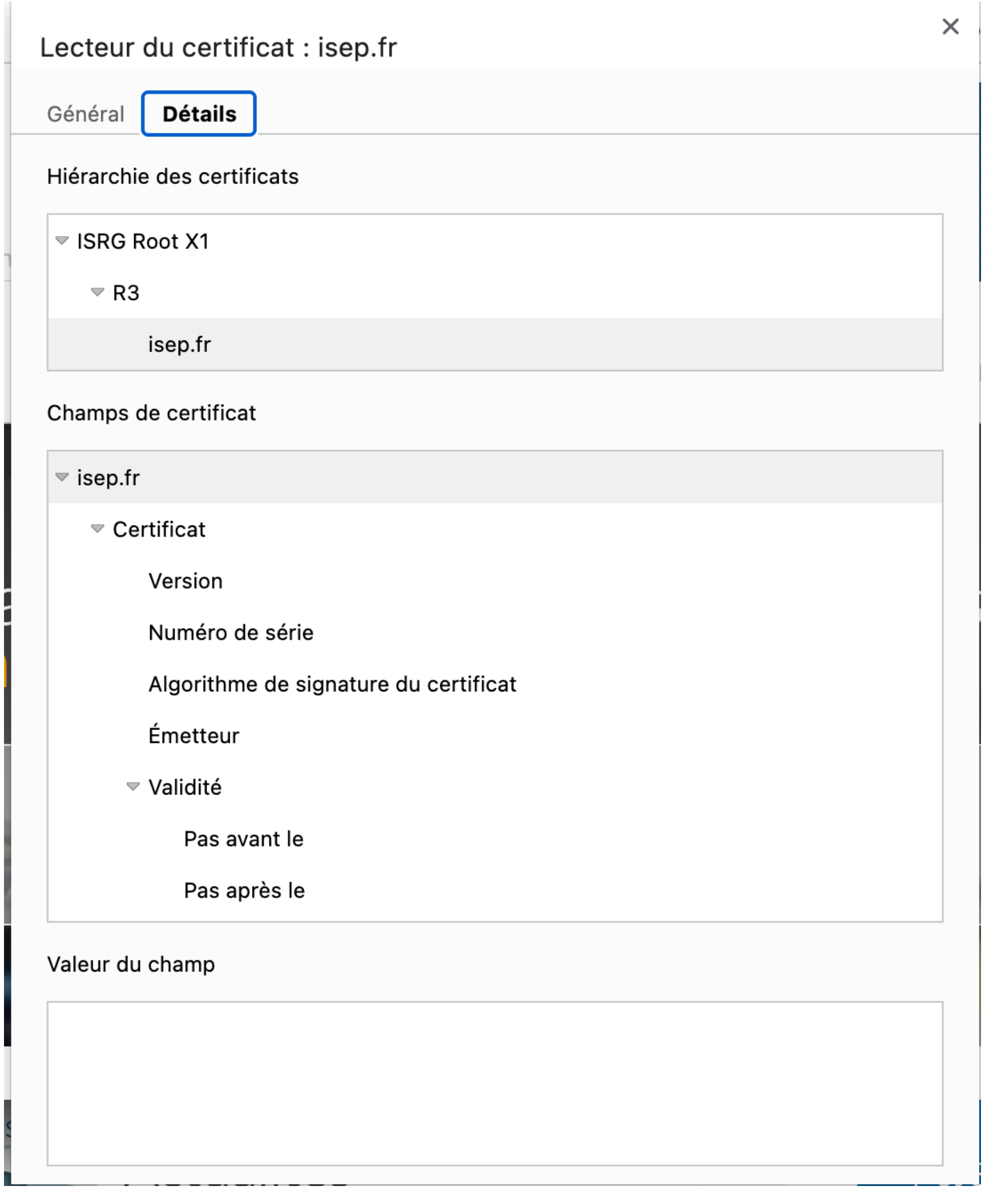
Durée de validité

Émis le	samedi 8 octobre 2022 à 01:14:27
Expire le	vendredi 6 janvier 2023 à 00:14:26

Empreintes

Empreinte SHA-256	C5 A9 C5 AE 13 BE 75 0B C1 07 D5 5A AB 01 AC 11 42 81 61 8D 51 9B C3 32 D0 18 DB 97 37 A5 99 F6
Empreinte SHA-1	AA D0 77 A9 D1 B2 89 F8 2F 54 F8 3C 7C 28 5D 0C A8 E1 67 BF

Example: https://www.isep.fr



Lecteur du certificat : isep.fr

Hiérarchie des certificats

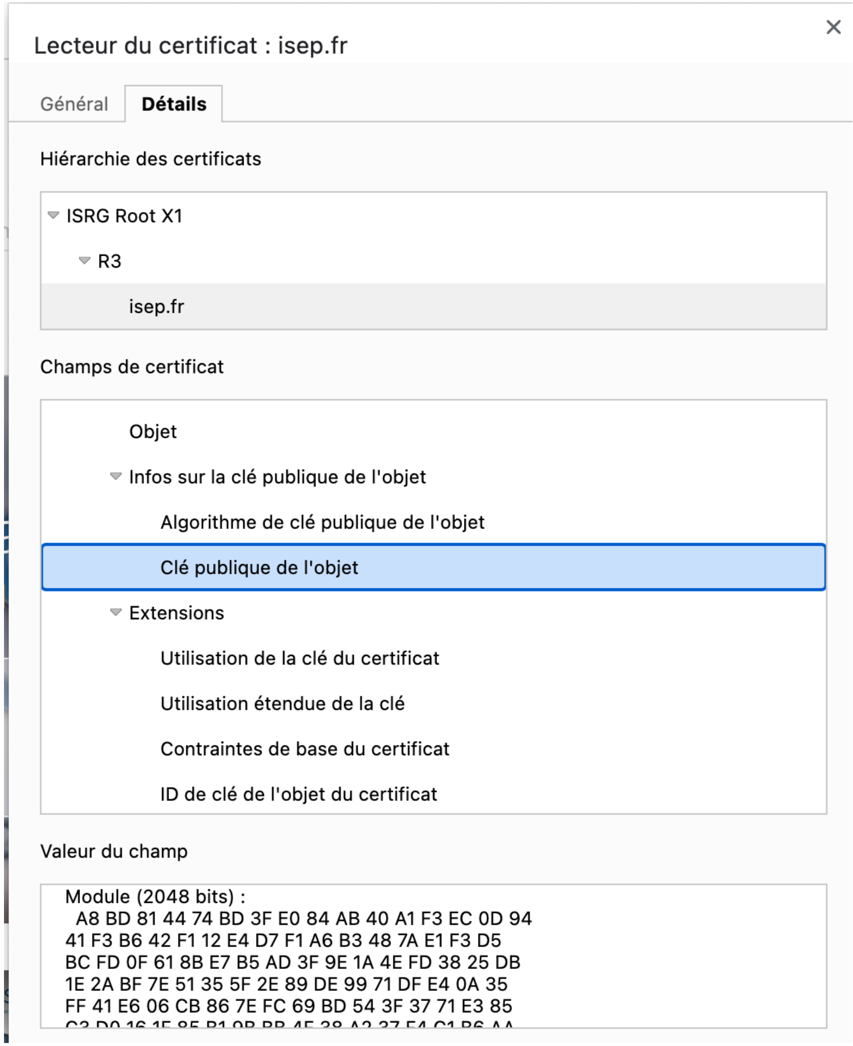
- ISRG Root X1
  - R3
    - isep.fr

Champs de certificat

- isep.fr
  - Certificat
    - Version
    - Numéro de série
    - Algorithme de signature du certificat
    - Émetteur
    - Validité
      - Pas avant le
      - Pas après le

Valeur du champ

Example: https://www.isep.fr



Lecteur du certificat : isep.fr

Hiérarchie des certificats

- ISRG Root X1
  - R3
    - isep.fr

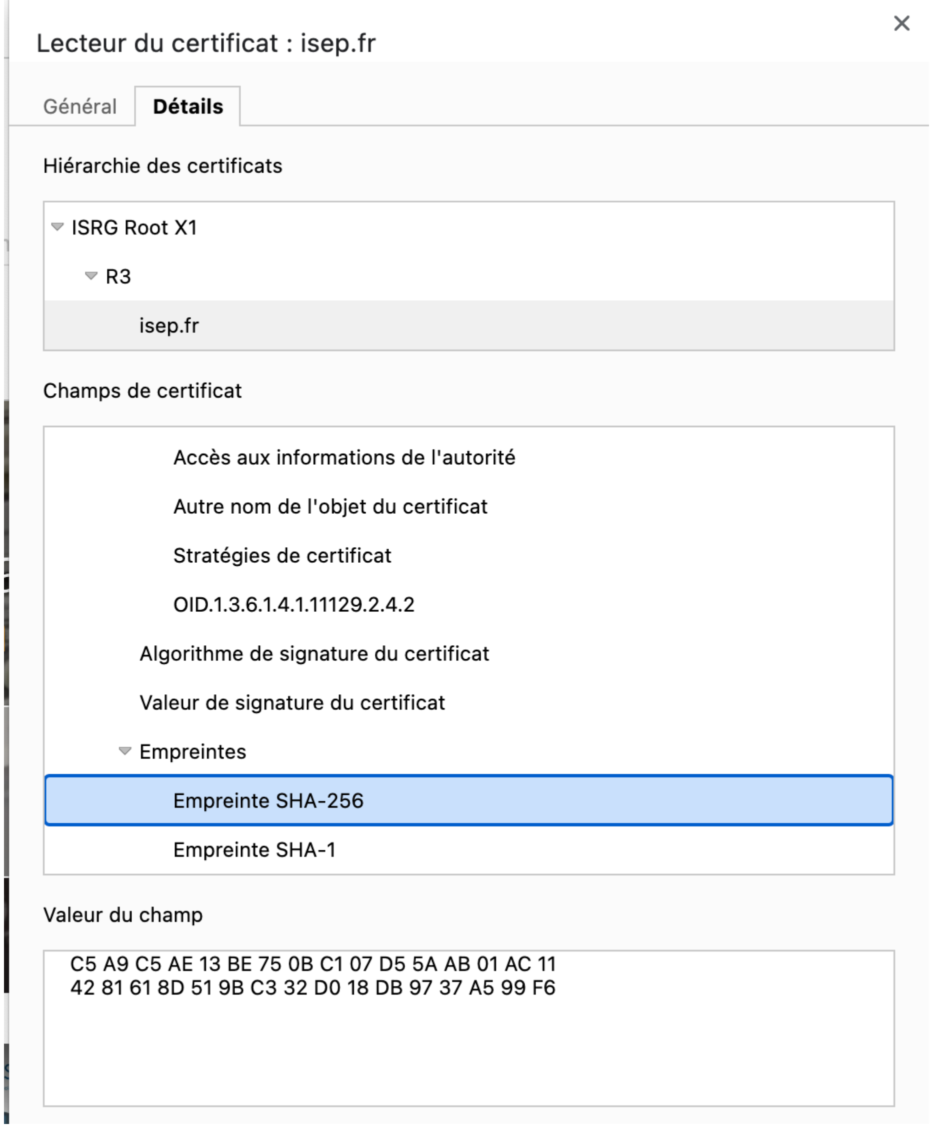
Champs de certificat

- Objet
- Infos sur la clé publique de l'objet
  - Algorithme de clé publique de l'objet
  - Clé publique de l'objet
- Extensions
  - Utilisation de la clé du certificat
  - Utilisation étendue de la clé
  - Contraintes de base du certificat
  - ID de clé de l'objet du certificat

Valeur du champ

```
Module (2048 bits) :
  A8 BD 81 44 74 BD 3F E0 84 AB 40 A1 F3 EC 0D 94
41 F3 B6 42 F1 12 E4 D7 F1 A6 B3 48 7A E1 F3 D5
BC FD 0F 61 8B E7 B5 AD 3F 9E 1A 4E FD 38 25 DB
1E 2A BF 7E 51 35 5F 2E 89 DE 99 71 DF E4 0A 35
FF 41 E6 06 CB 86 7E FC 69 BD 54 3F 37 71 E3 85
C3 D0 16 1F 85 B1 9B BB 4F 38 A2 37 F4 C1 B6 AA
```

Example: https://www.isep.fr



Lecteur du certificat : isep.fr

Hiérarchie des certificats

- ISRG Root X1
  - R3
    - isep.fr

Champs de certificat

- Accès aux informations de l'autorité
- Autre nom de l'objet du certificat
- Stratégies de certificat
- OID.1.3.6.1.4.1.11129.2.4.2
- Algorithme de signature du certificat
- Valeur de signature du certificat
- Empreintes
  - Empreinte SHA-256
  - Empreinte SHA-1

Valeur du champ

```
C5 A9 C5 AE 13 BE 75 0B C1 07 D5 5A AB 01 AC 11
42 81 61 8D 51 9B C3 32 D0 18 DB 97 37 A5 99 F6
```

Thanks !

Nour EL MADHOUN

Associate Professor

nour.el-madhoun@isep.fr