Cybersecurity Course

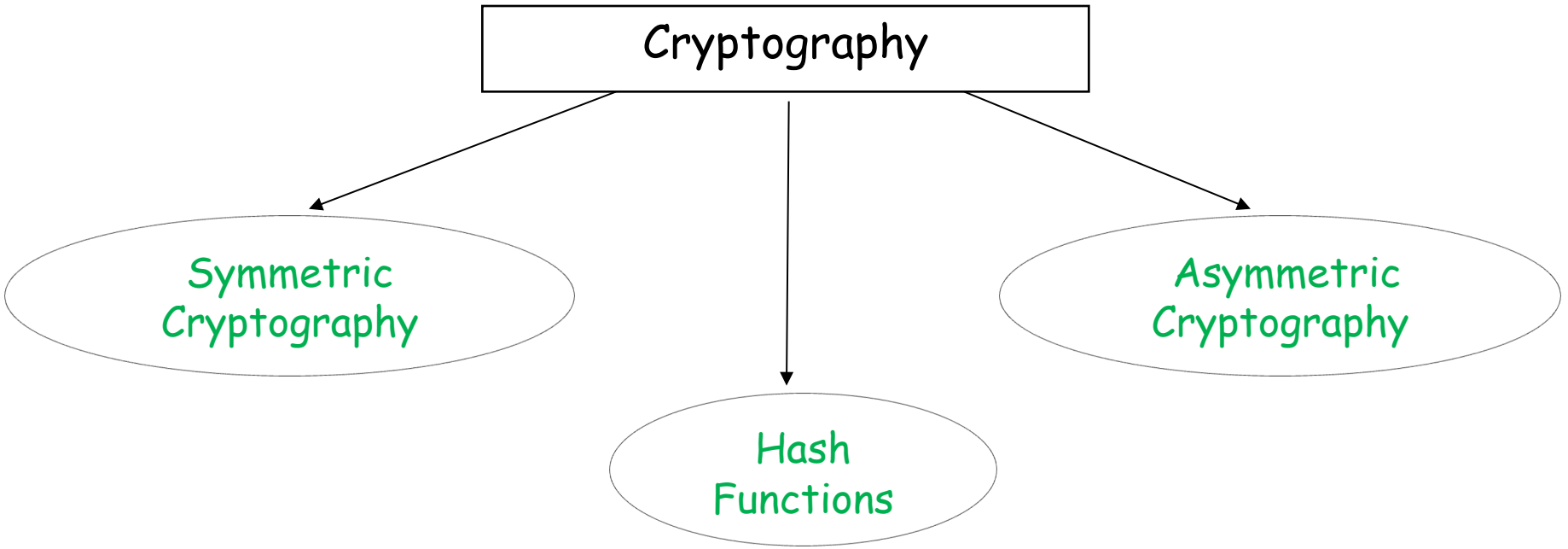
Lecture 2: Introduction to Cryptography

Nour EL MADHOUN

Associate Professor

nour.el-madhoun@isep.fr

Office: L219


Cryptography
Symmetric Cryptography
Hash Functions
Asymmetric Cryptography

Cryptography

Symmetric Cryptography → Confidentiality of the message

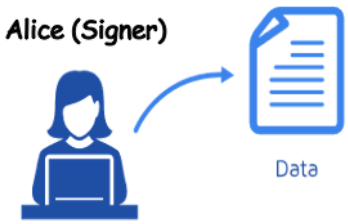
Asymmetric Cryptography → Confidentiality of the message

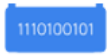
Hash Functions → Integrity of the message

Cryptography

Asymmetric Cryptography
+
Hash Function

Electronic Signature





Electronic signature

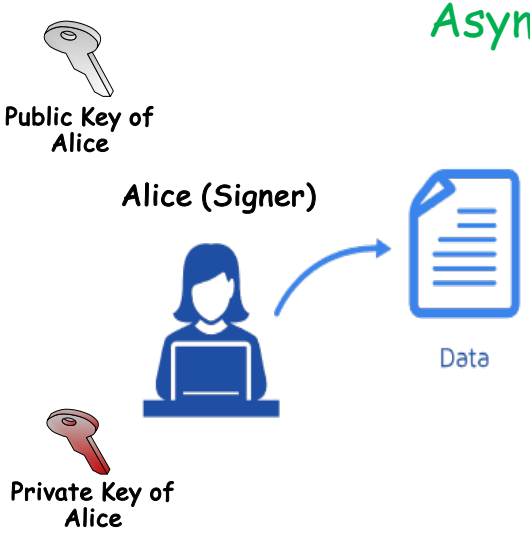
Authentication of the Signer

Non-repudiation for the Signer

Integrity of the Data

Electronic Signature

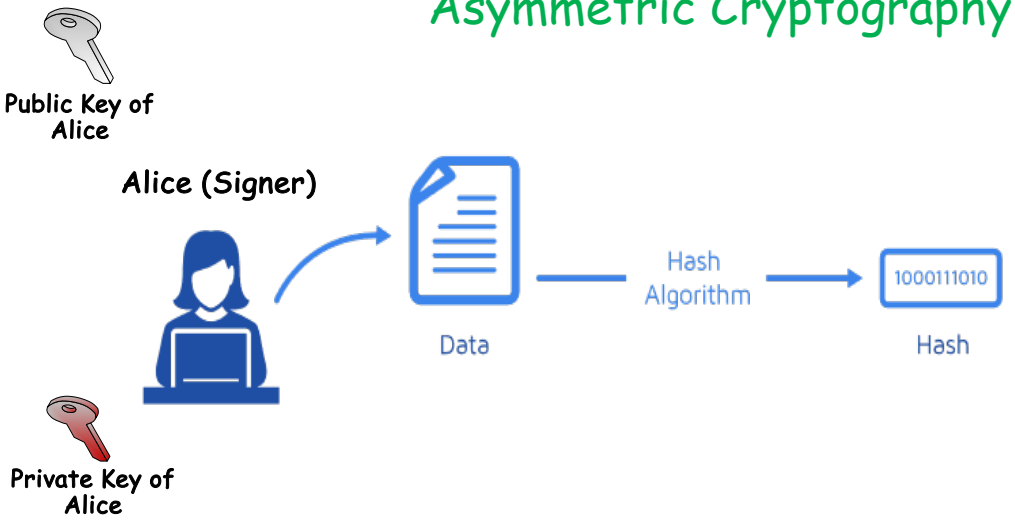
Asymmetric Cryptography + Hash Function



Bob (Receiver)

Electronic Signature

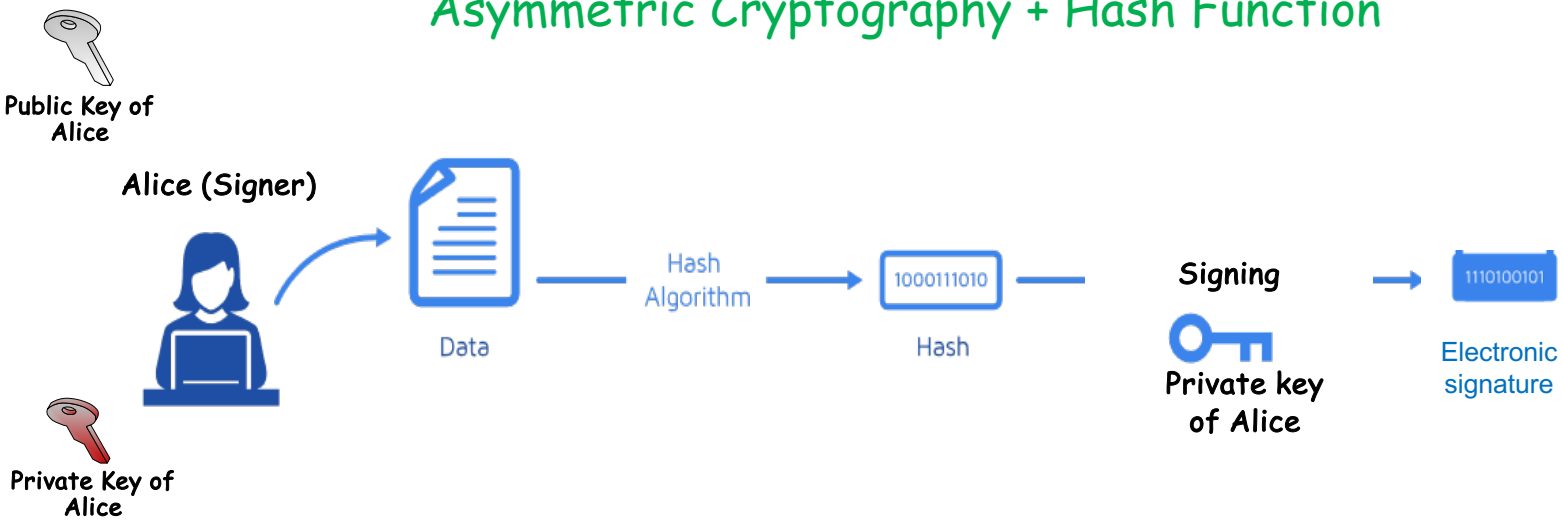
Asymmetric Cryptography + Hash Function



Bob (Receiver)

Electronic Signature

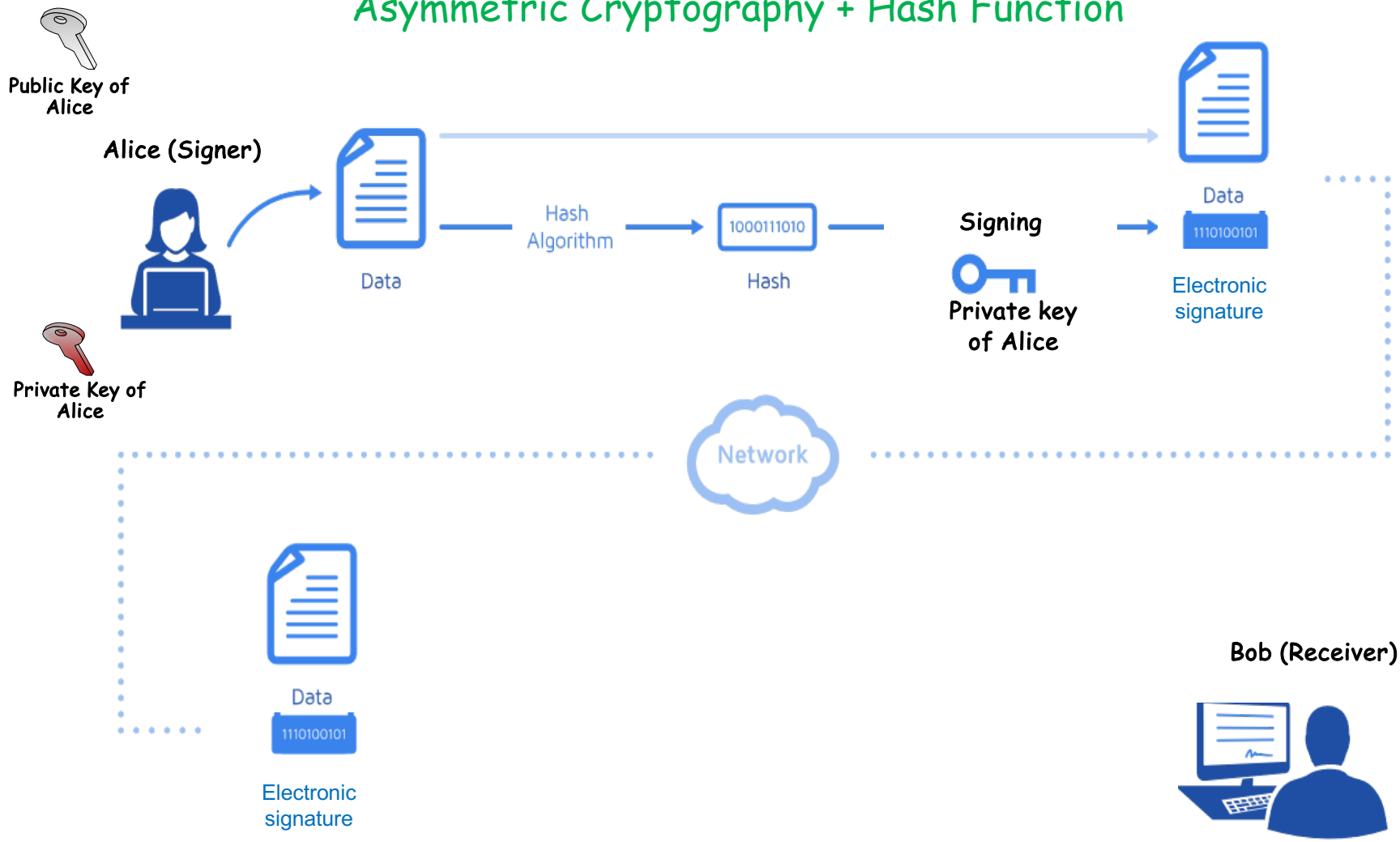
Asymmetric Cryptography + Hash Function



Bob (Receiver)

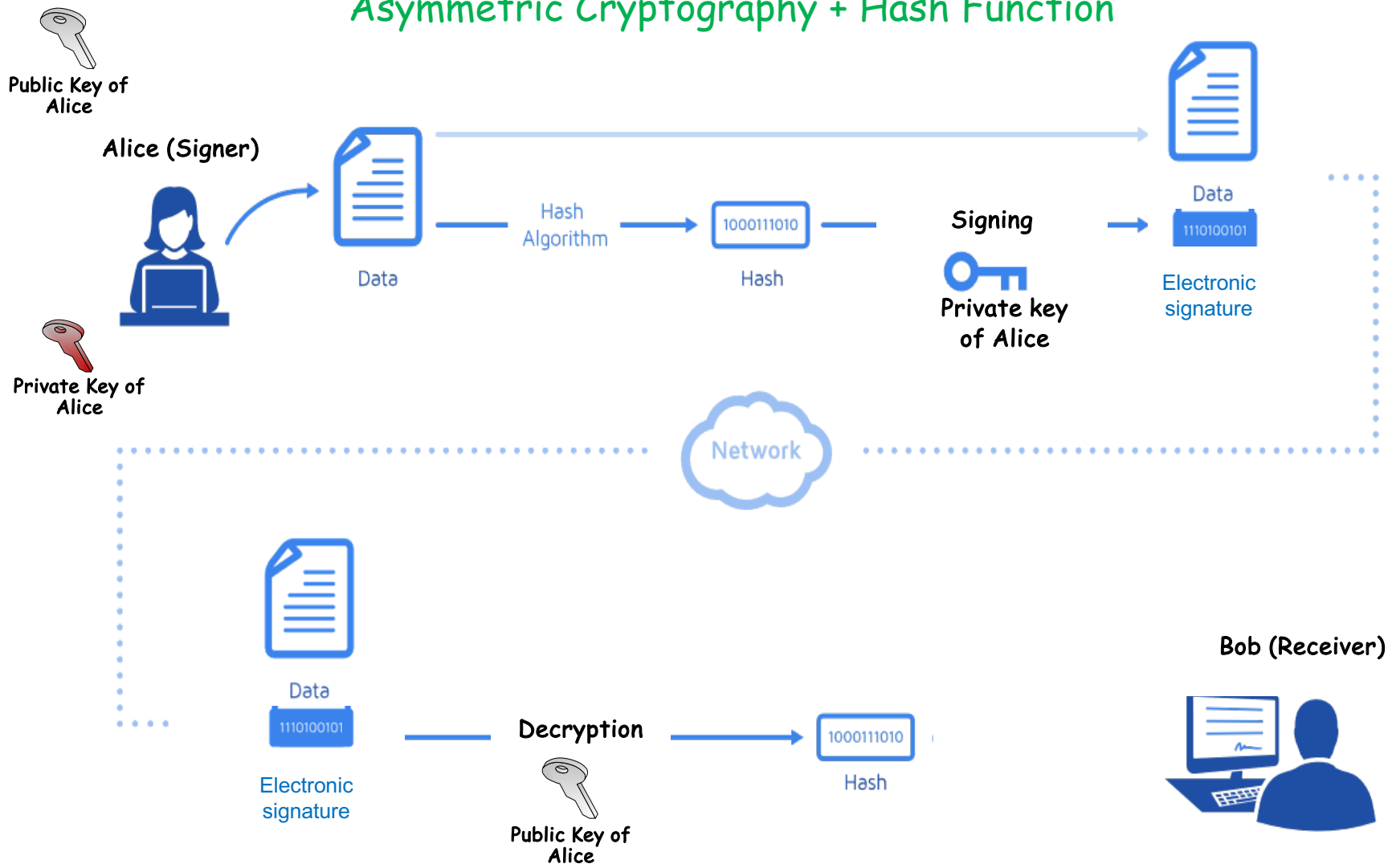
Electronic Signature

Asymmetric Cryptography + Hash Function



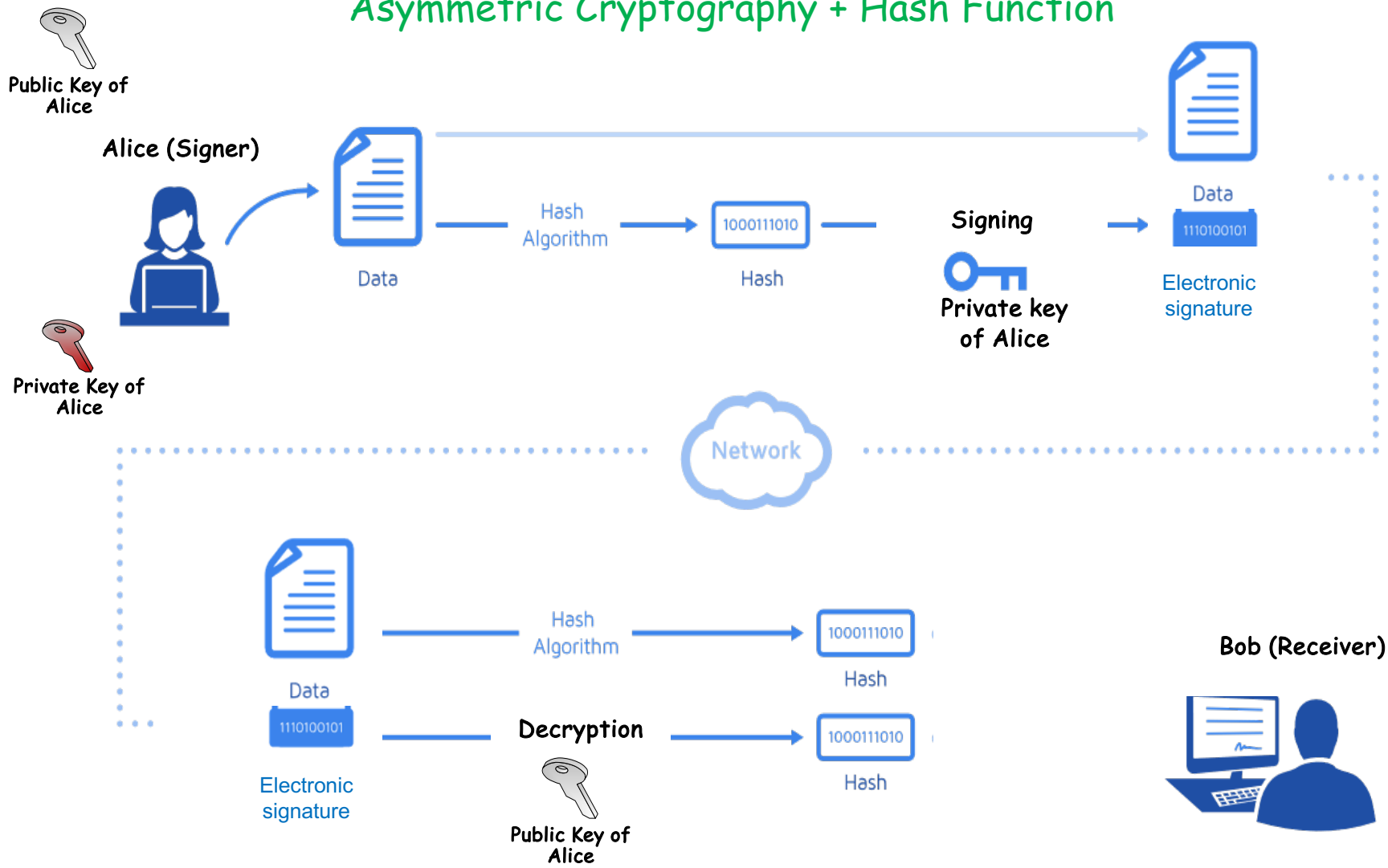
Electronic Signature

Asymmetric Cryptography + Hash Function



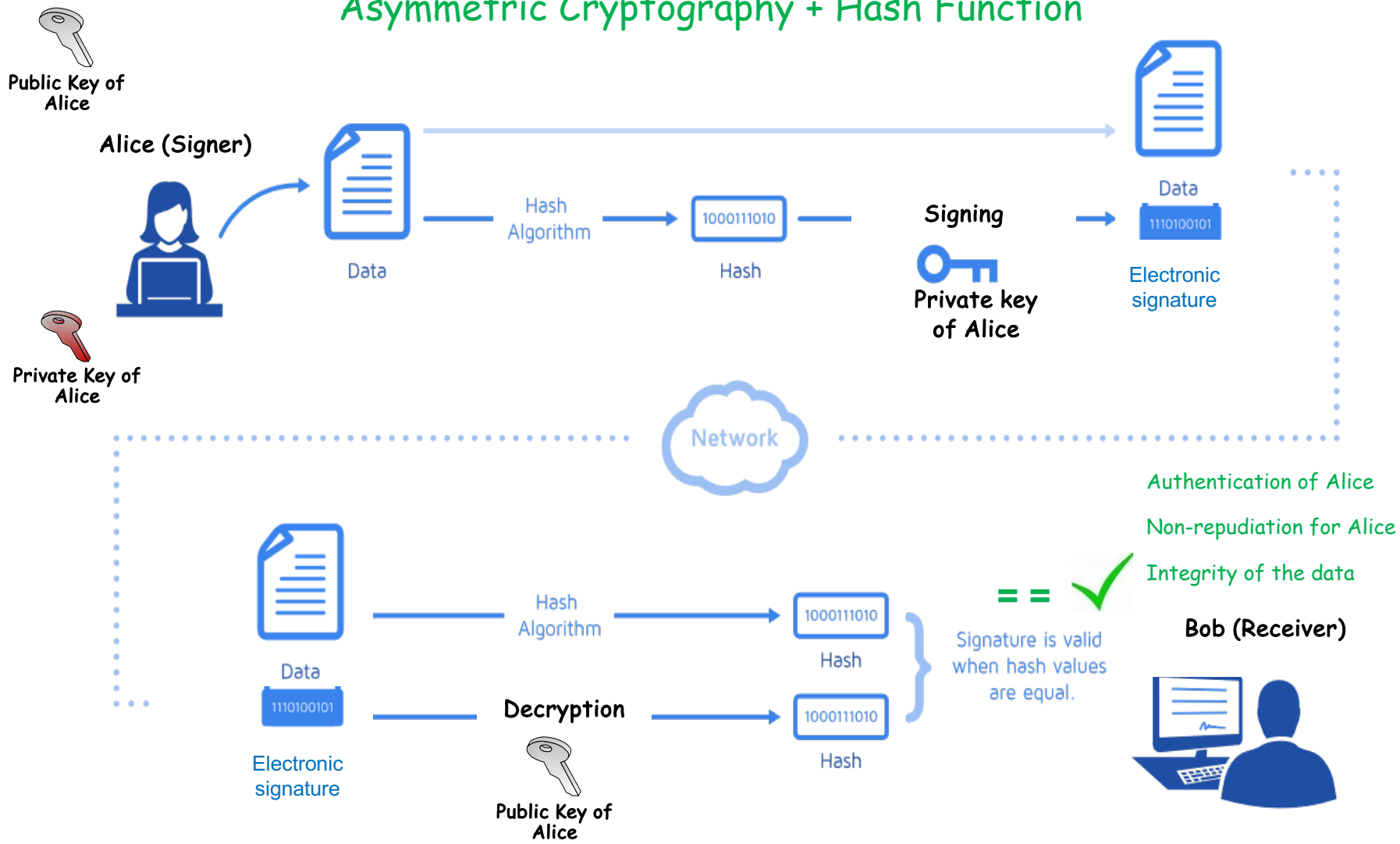
Electronic Signature

Asymmetric Cryptography + Hash Function



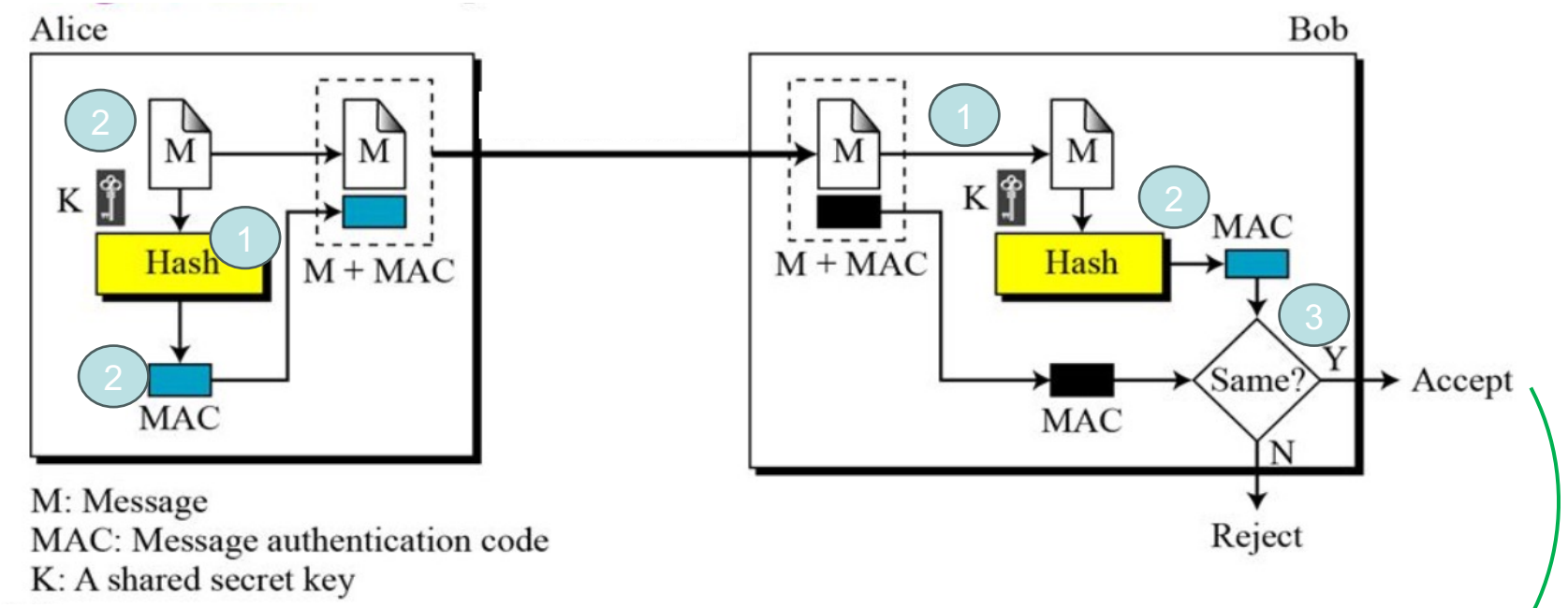
Electronic Signature

Asymmetric Cryptography + Hash Function



MAC (Message Authentication Code)

Symmetric Cryptography + Hash Function

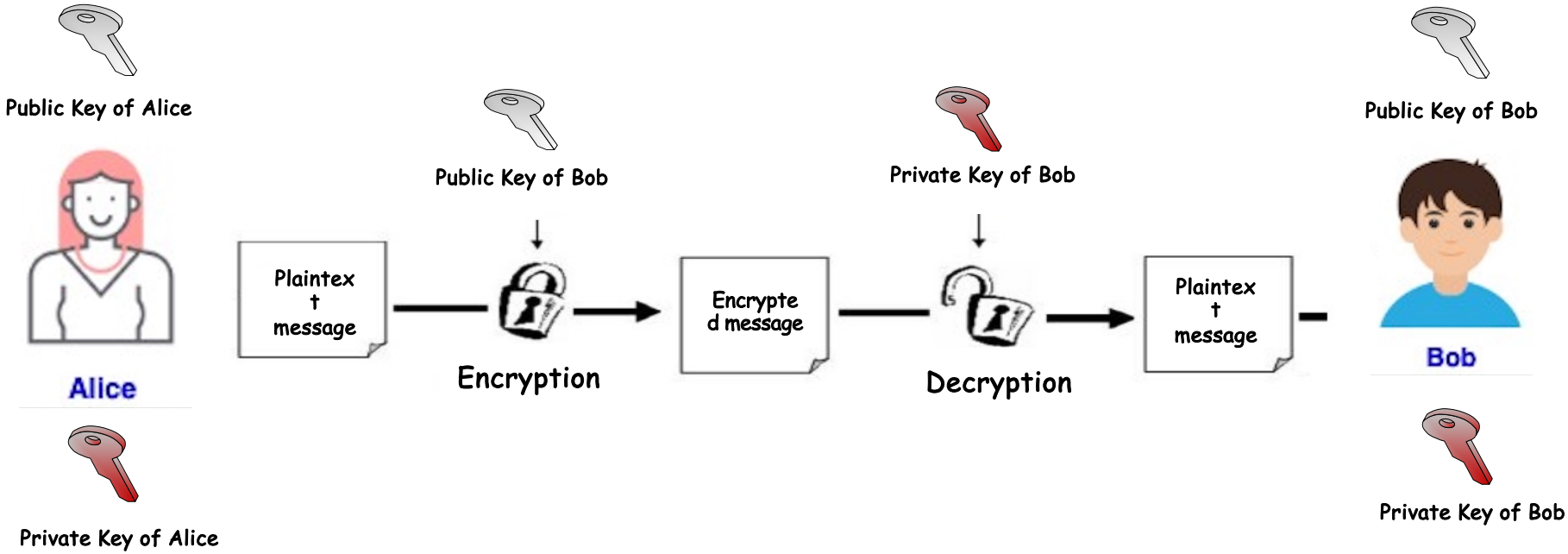


- Authentication of the Signer
- Integrity of the data

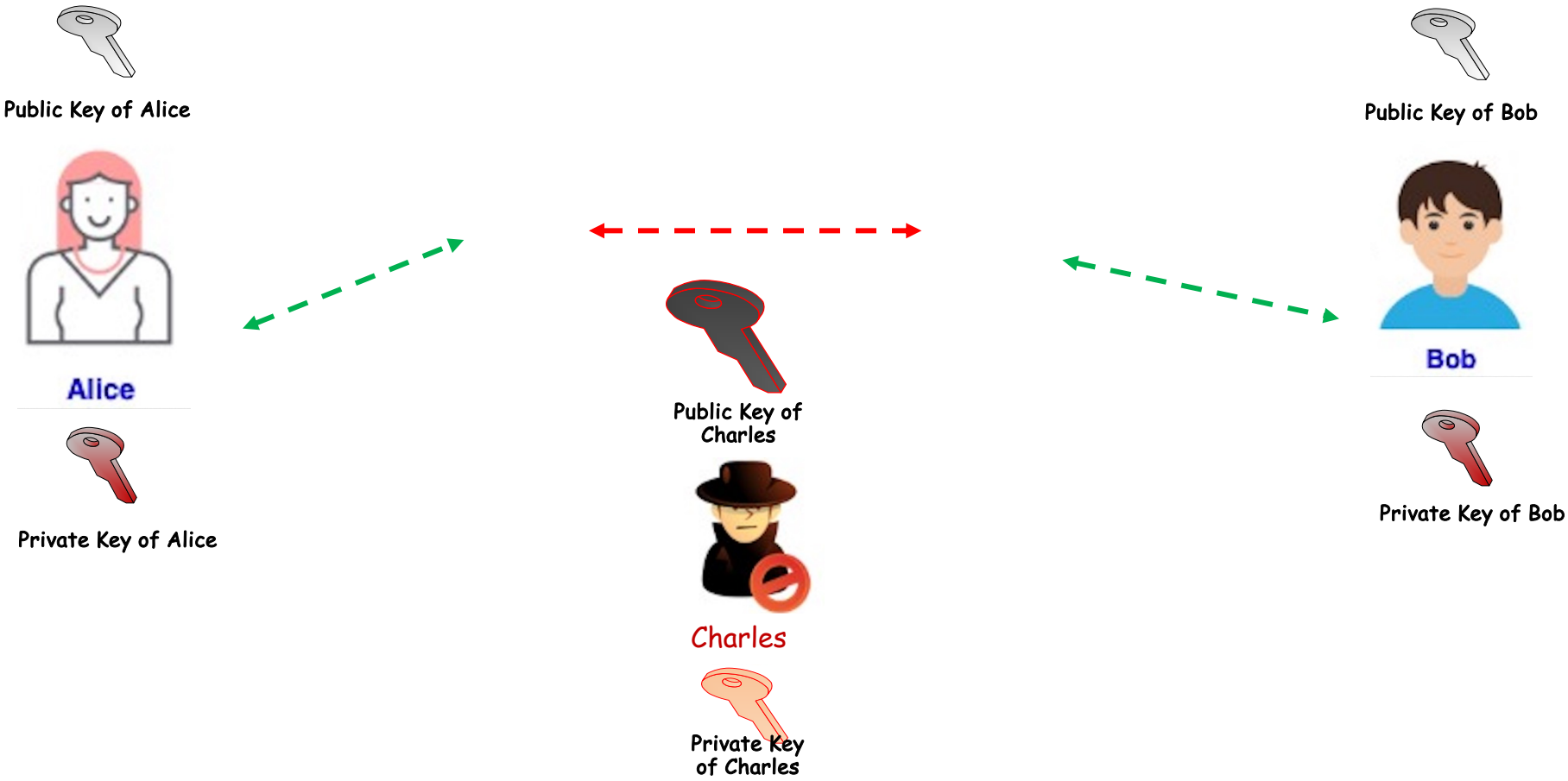
Electronic Certificate

Man in the Middle Attack (MITM)

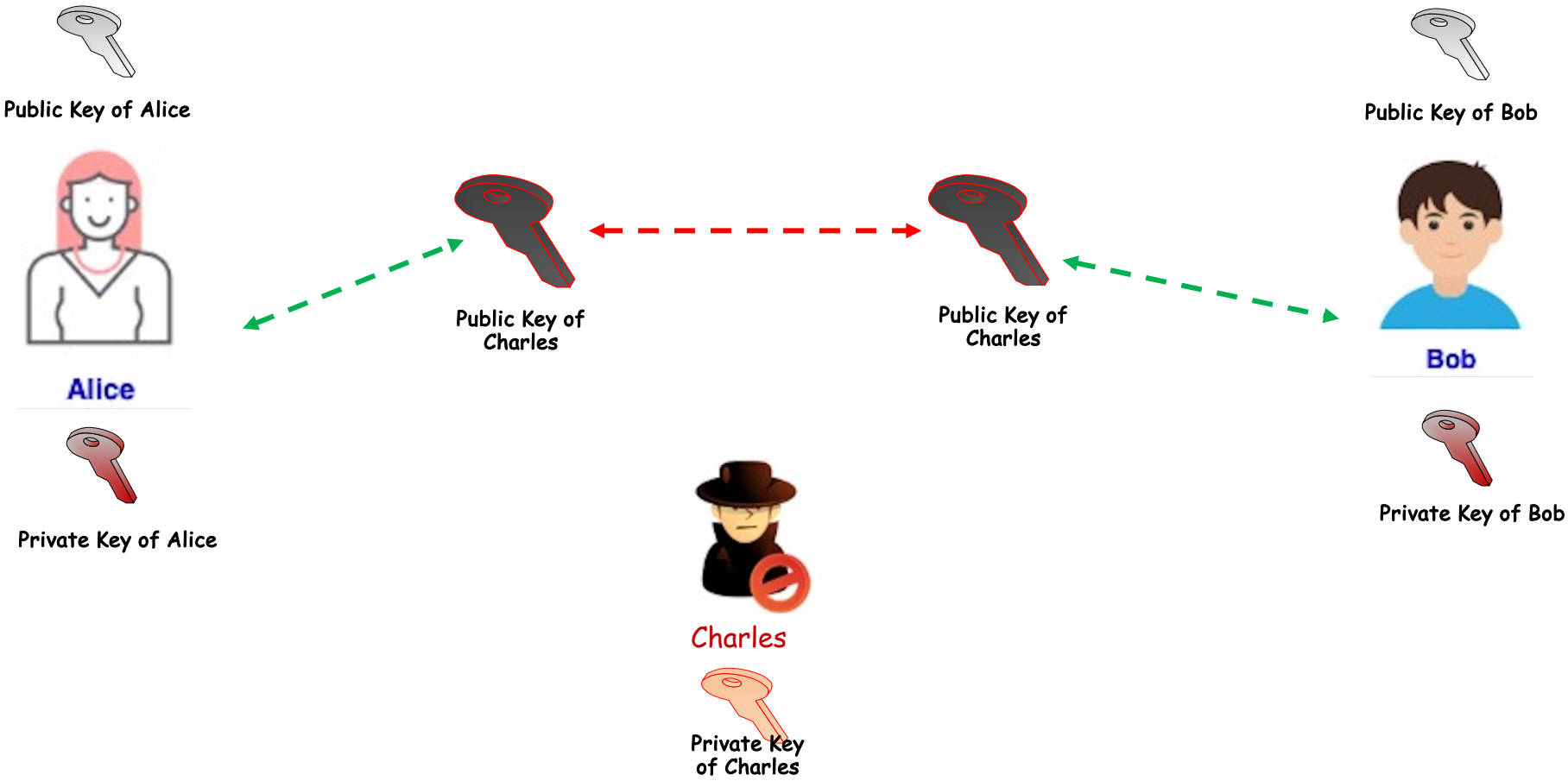
How to guarantee the identity of Bob linked to his public key?



Man in the Middle Attack (MITM)



Man in the Middle Attack (MITM)



Man in the Middle Attack (MITM)

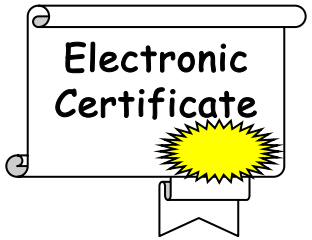
Public Key of Alice

Alice

Private Key of Alice

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

Solution

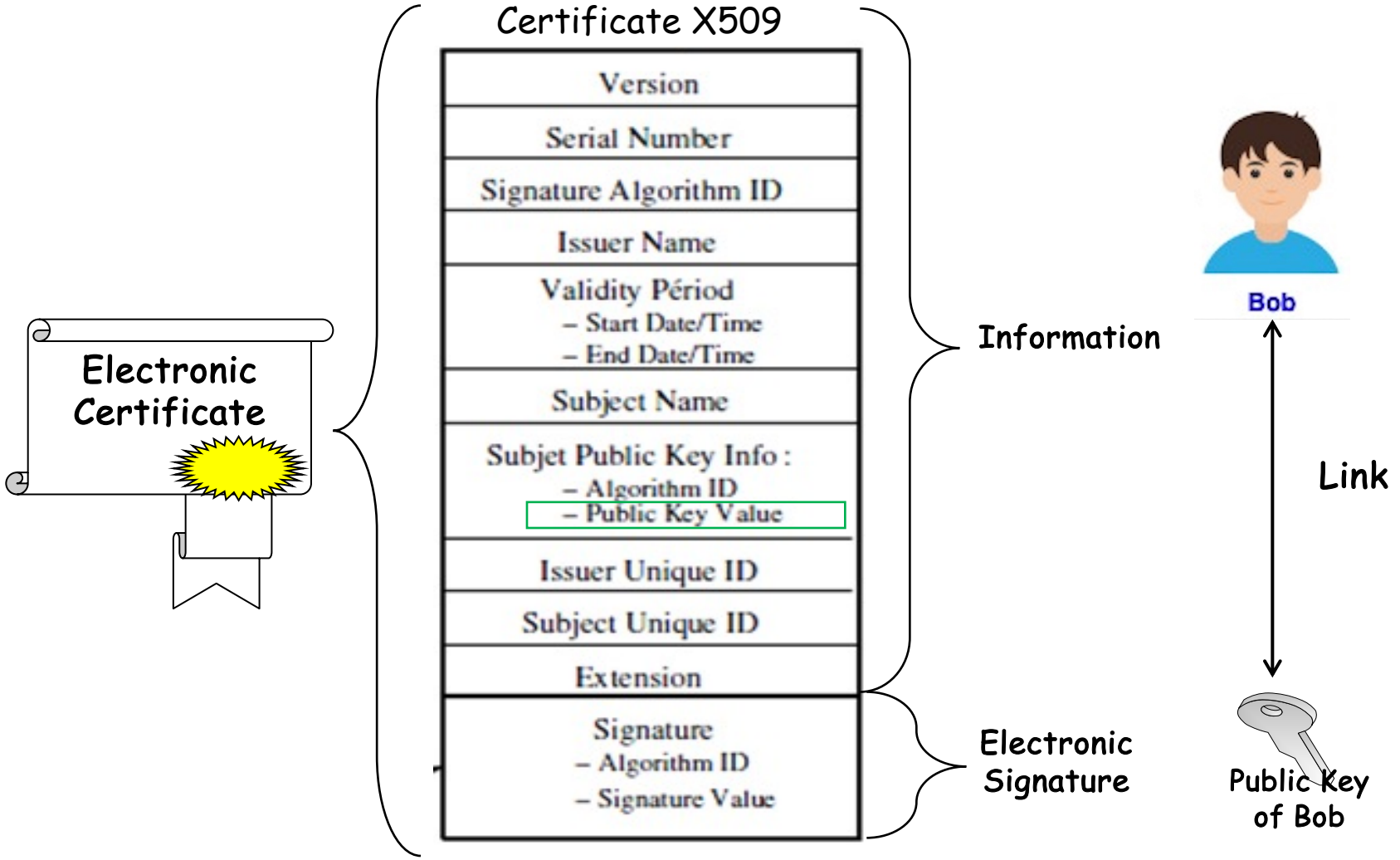


Public Key of Bob

Bob

Private Key of Bob

Electronic Certificate



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

2

Private Key of CA

2 Signing

Information

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

1 Hash Function

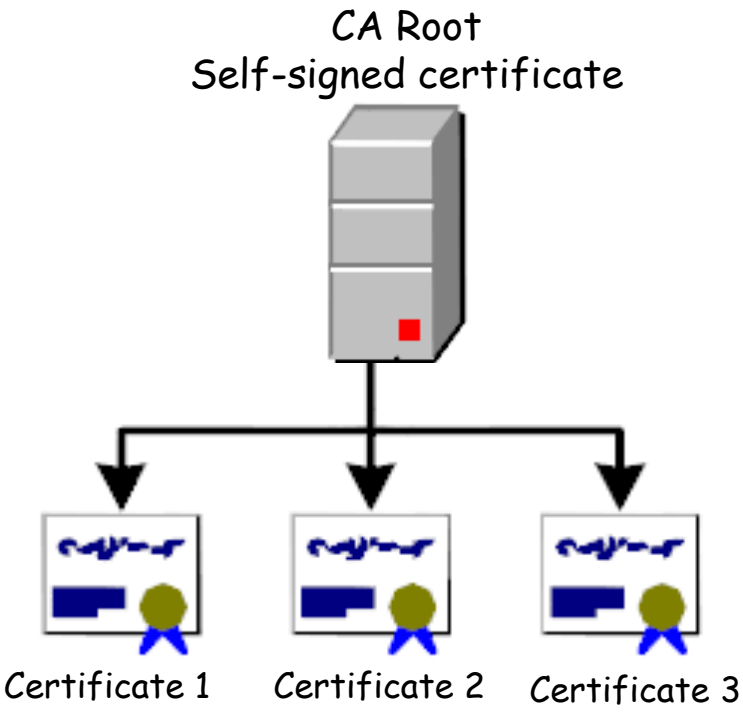
Hash of the information

Certificate X509

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

Trust Models

1- Root CA Model

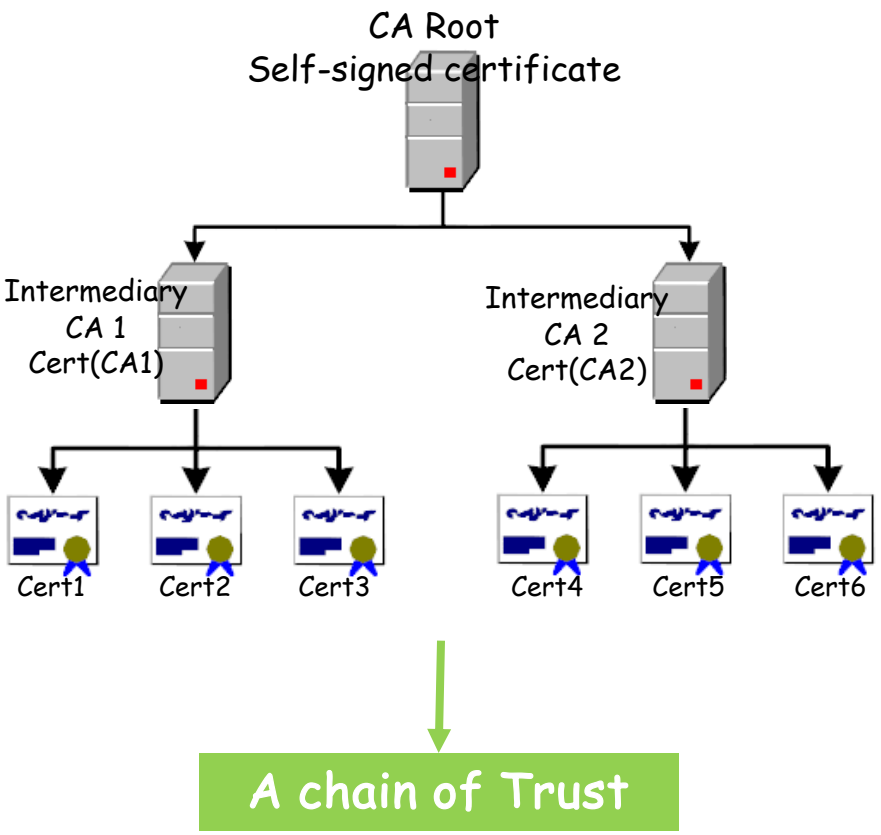


1- The CA root has its private/public keys and its self-signed certificate which contains its public key. The CA signs by using its private key the certificates 1, 2, 3

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



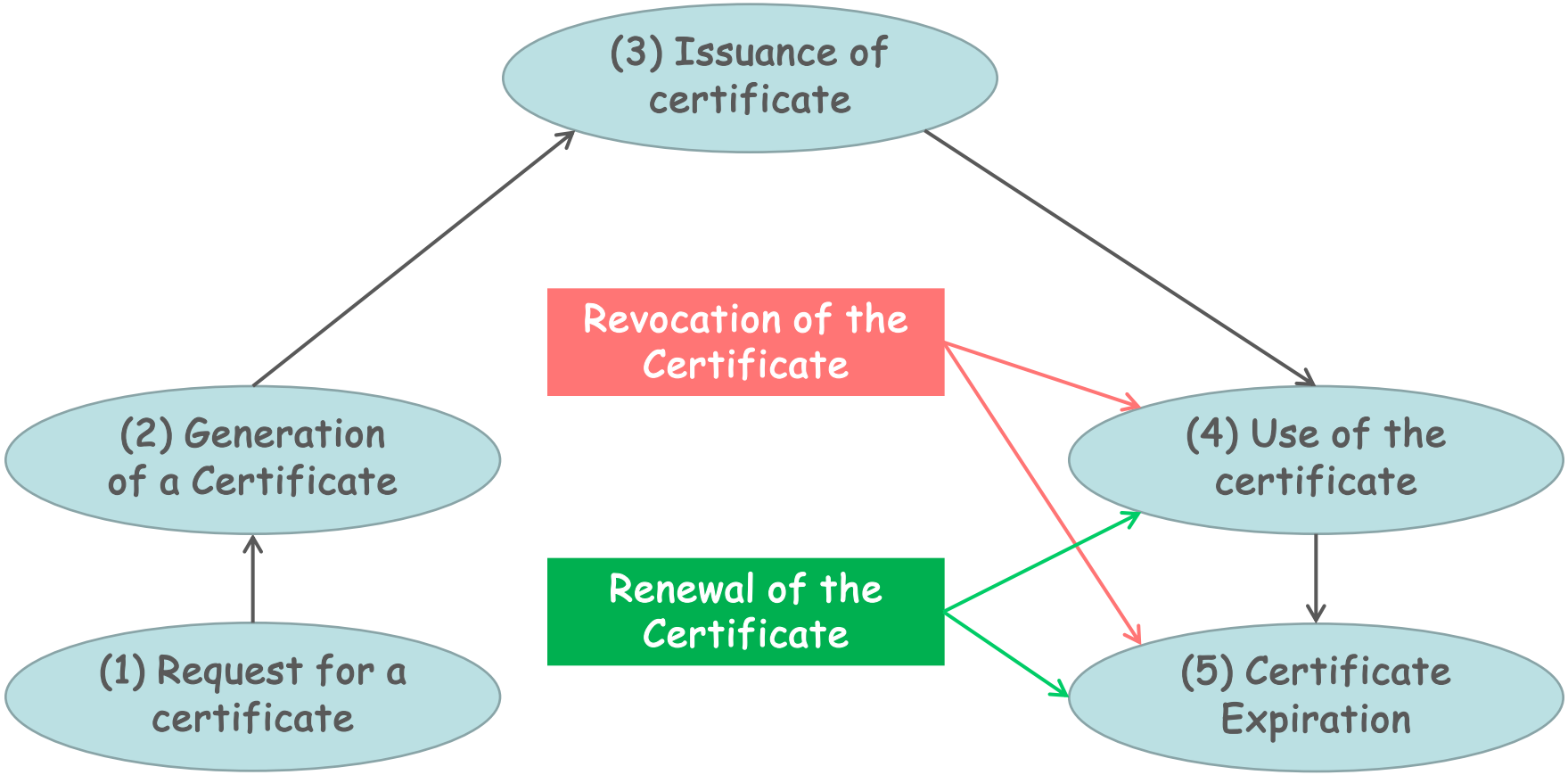
1- The CA root has its private/public keys and its self-signed certificate which contains its public key.

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.

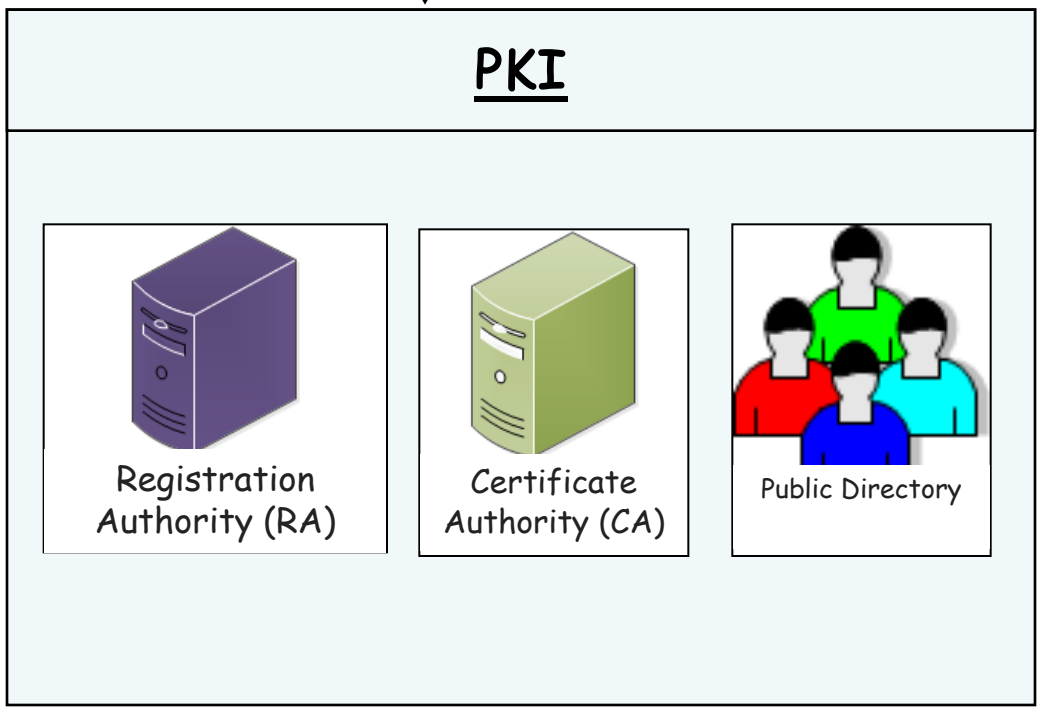
The life cycle of an electronic certificate



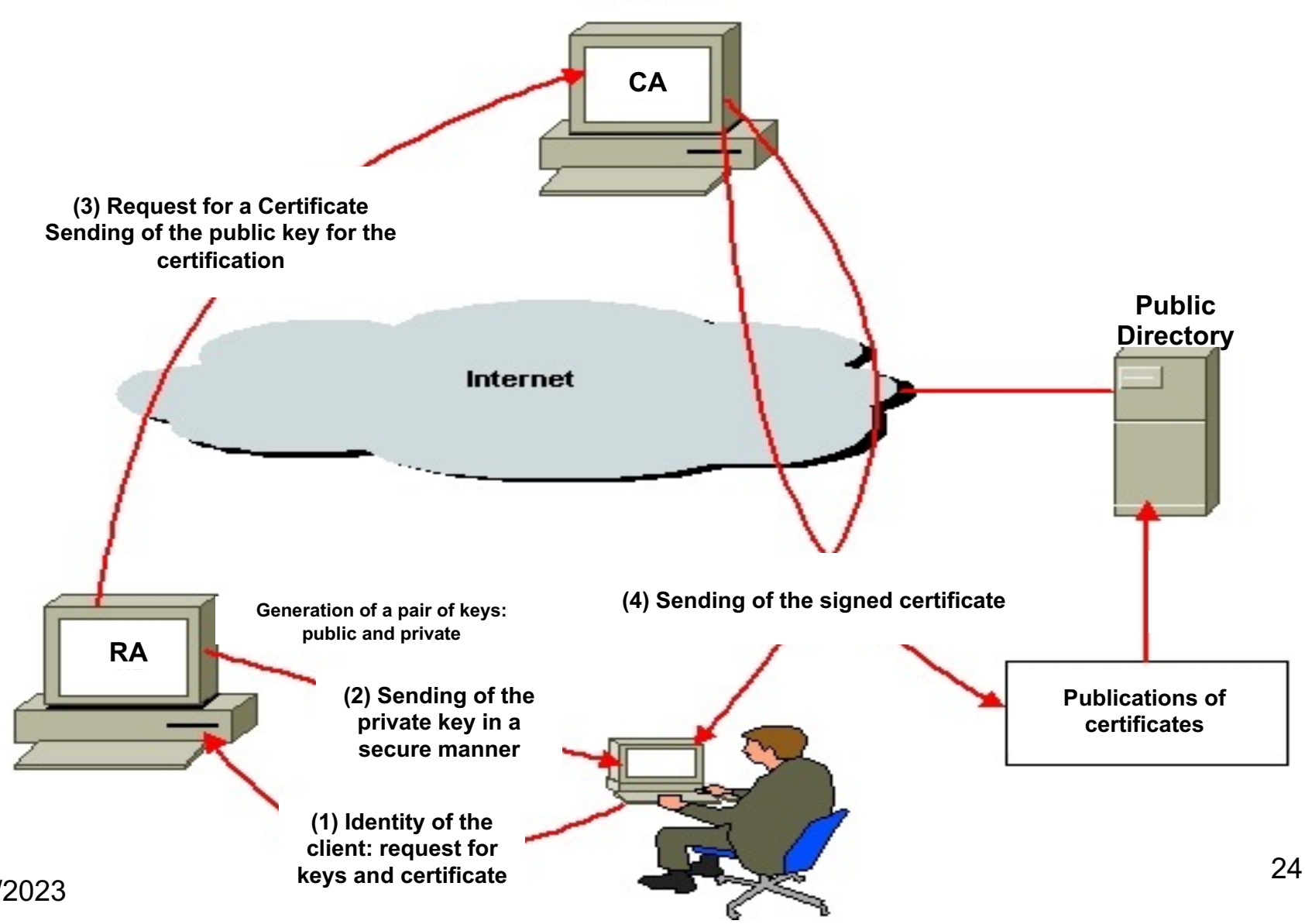
PKI (Public Key Infrastructure)

The life cycle of an electronic certificate

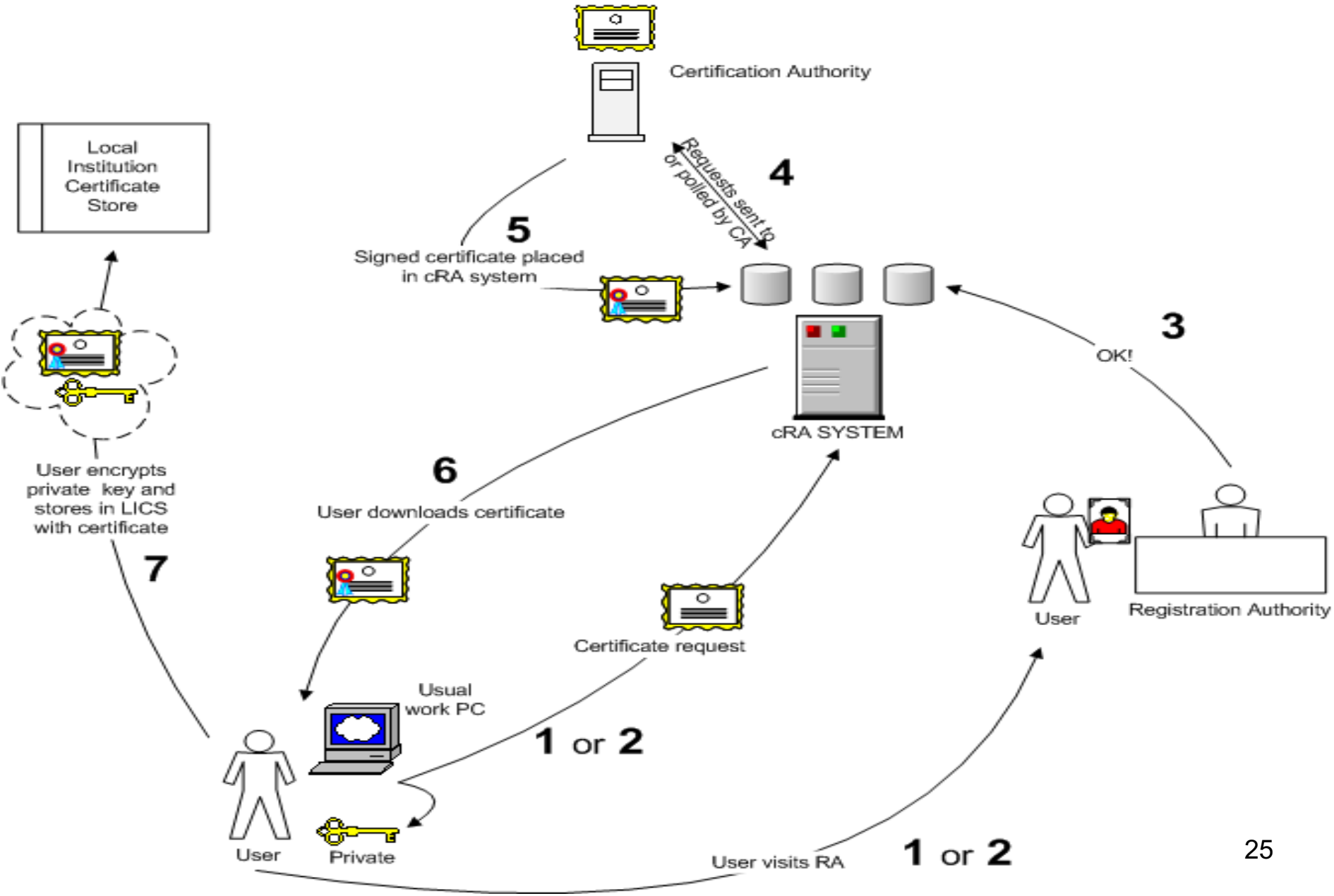
Life Cycle Management of Certificates is ensured:



PKI (Public Key Infrastructure)

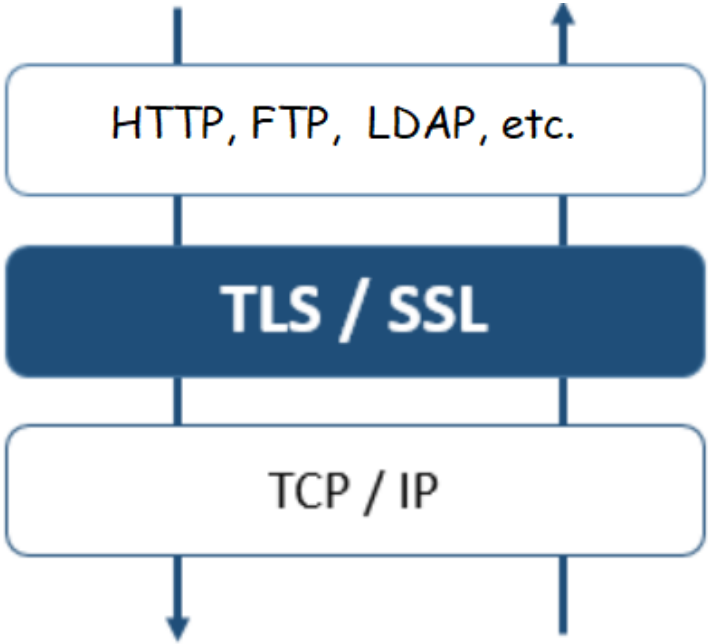


PKI (Public Key Infrastructure)



TLS Protocol

TLS Protocol



- SSL/TLS: Protocol aims to secure communications between a Client and a Server
- SSL: Secure Socket Layer (Version 1)
- TLS: Transport Layer Security
- TLS: Version 3 of the SSL protocol

TLS Protocol

Client

TLS operates in a client-server mode

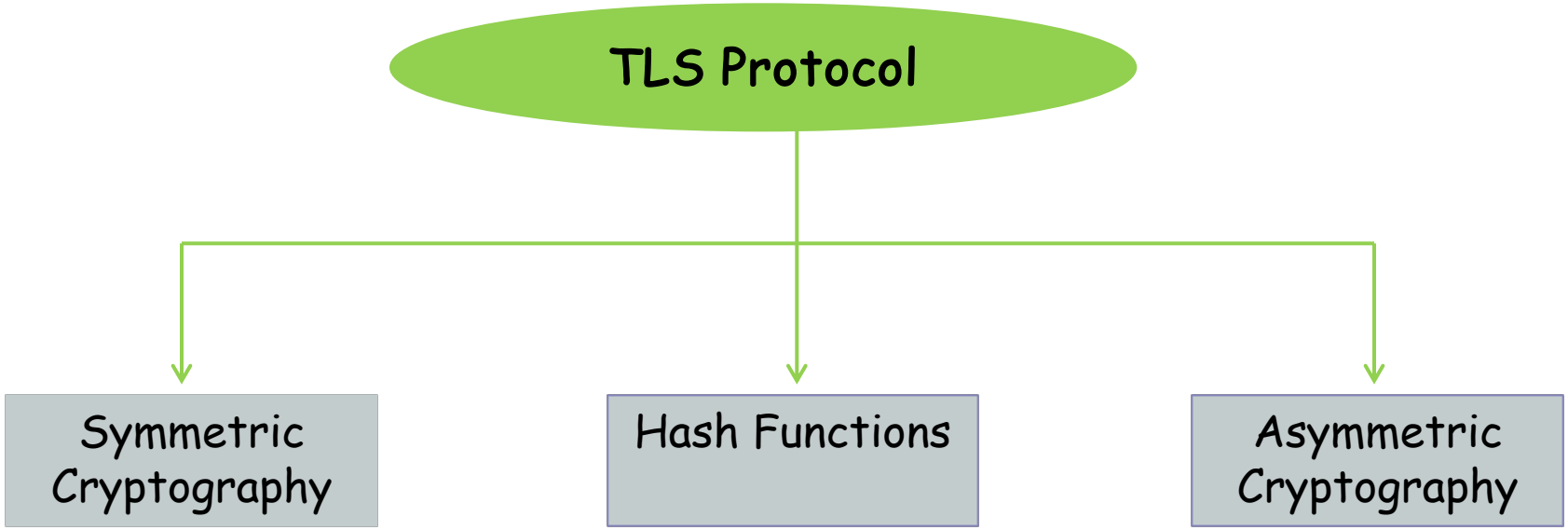
Allows to ensure the security properties :

* Server authentication
* Server non-repudiation
* Confidentiality of exchanged data

Server

Using Cryptography

TLS Protocol



TLS Protocol

Example Web Application

Browser

Web Server

TLS Protocol

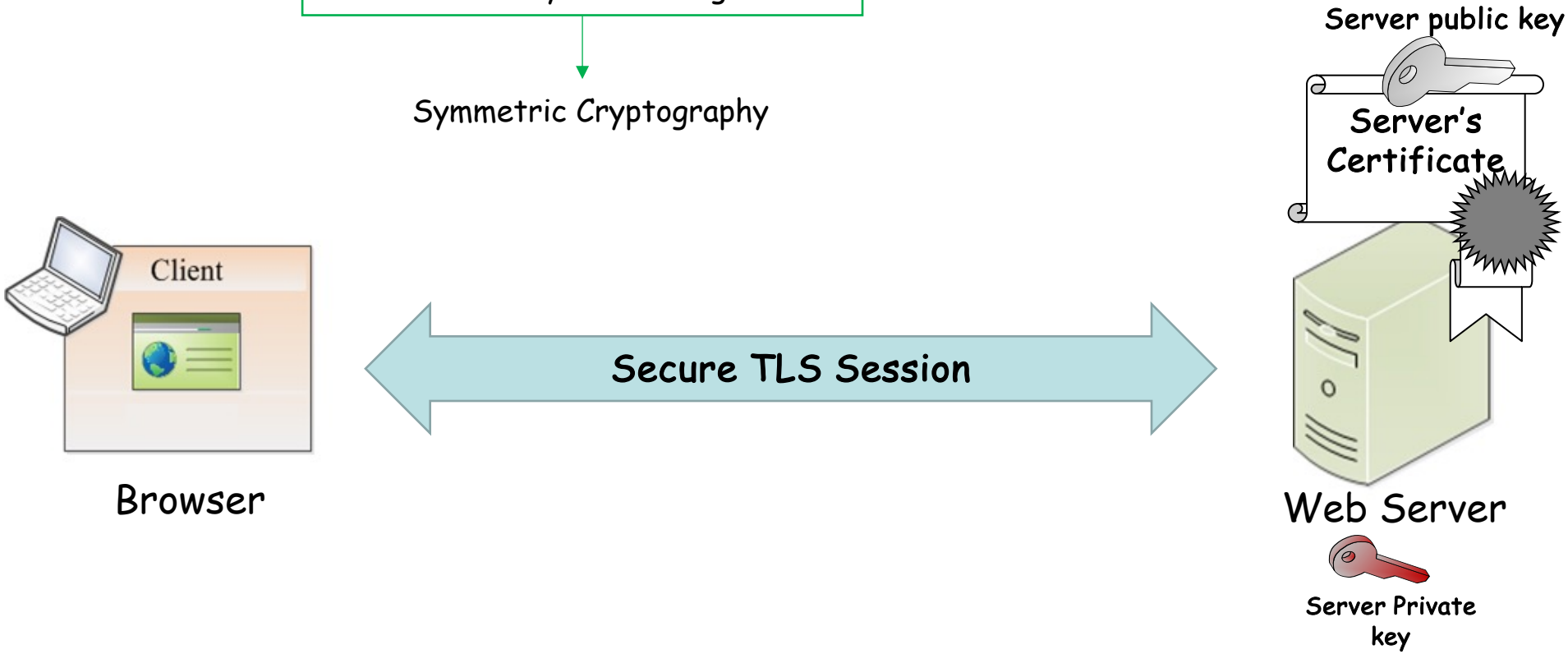
TLS Allows to ensure the security properties :

* Server authentication
* Server non-repudiation

→ Electronic Signature

* Confidentiality of exchanged data

→ Symmetric Cryptography

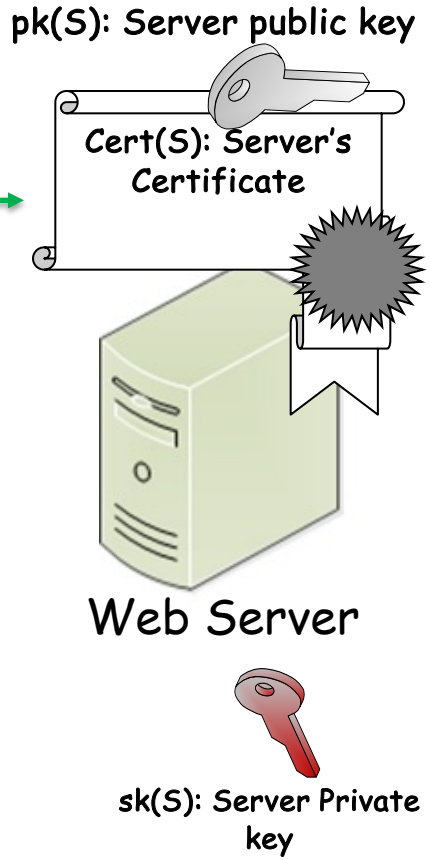


TLS Protocol

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

Generates

- pk(CA)/sk(CA): self-generated
- Cert(CA): self-Signed
- CA signs by using its sk(CA) the server's certificate Cert(S).



TLS Protocol

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

Generates

- pk(CA)/sk(CA): self-generated
- Cert(CA): self-Signed
- CA signs by using its sk(CA) the server's certificate Cert(S).

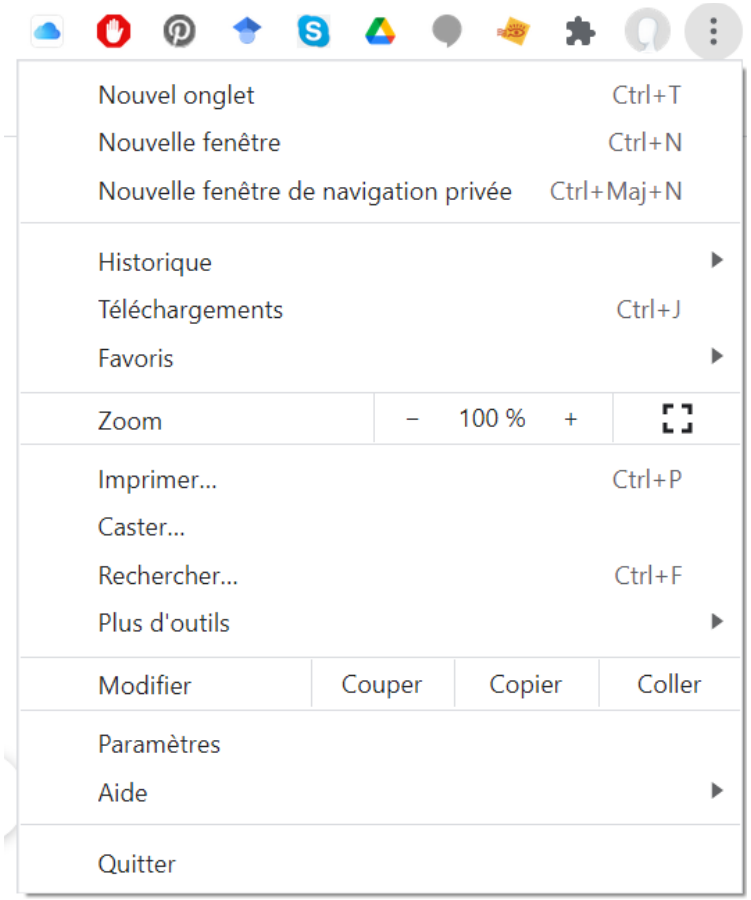
- The browser stores a list of CAs root and intermediary of trust (their certificates).

TLS Protocol

Browser

- The browser stores a list of CAs root of trust.

For example for "Google Chrome"

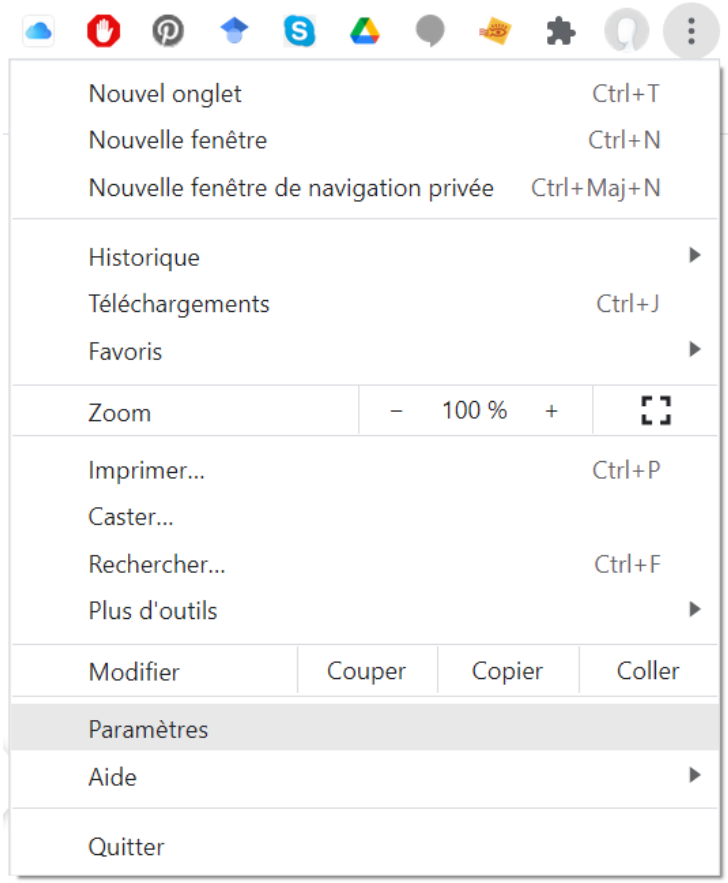


TLS Protocol

Browser

- The browser stores a list of CAs root of trust.

For example for "Google Chrome"



TLS Protocol

Browser

- The browser stores a list of CAs root of trust.

For example for "Google Chrome"

Confidentialité et sécurité

Effacer les données de navigation
Effacer l'historique, supprimer les cookies, vider le cache, etc.

Cookies et autres données des sites
Les cookies tiers sont bloqués lorsque vous utilisez le mode navigation privée

Sécurité
Navigation sécurisée (protection contre les sites dangereux) et autres paramètres de sécurité

Paramètres des sites
Permet de contrôler les informations que les sites peuvent utiliser et afficher (position, appareil photo, fenêtres pop-up et plus)

Privacy Sandbox
Les fonctionnalités à l'essai sont activées

TLS Protocol

Browser

- The browser stores a list of CAs root of trust.

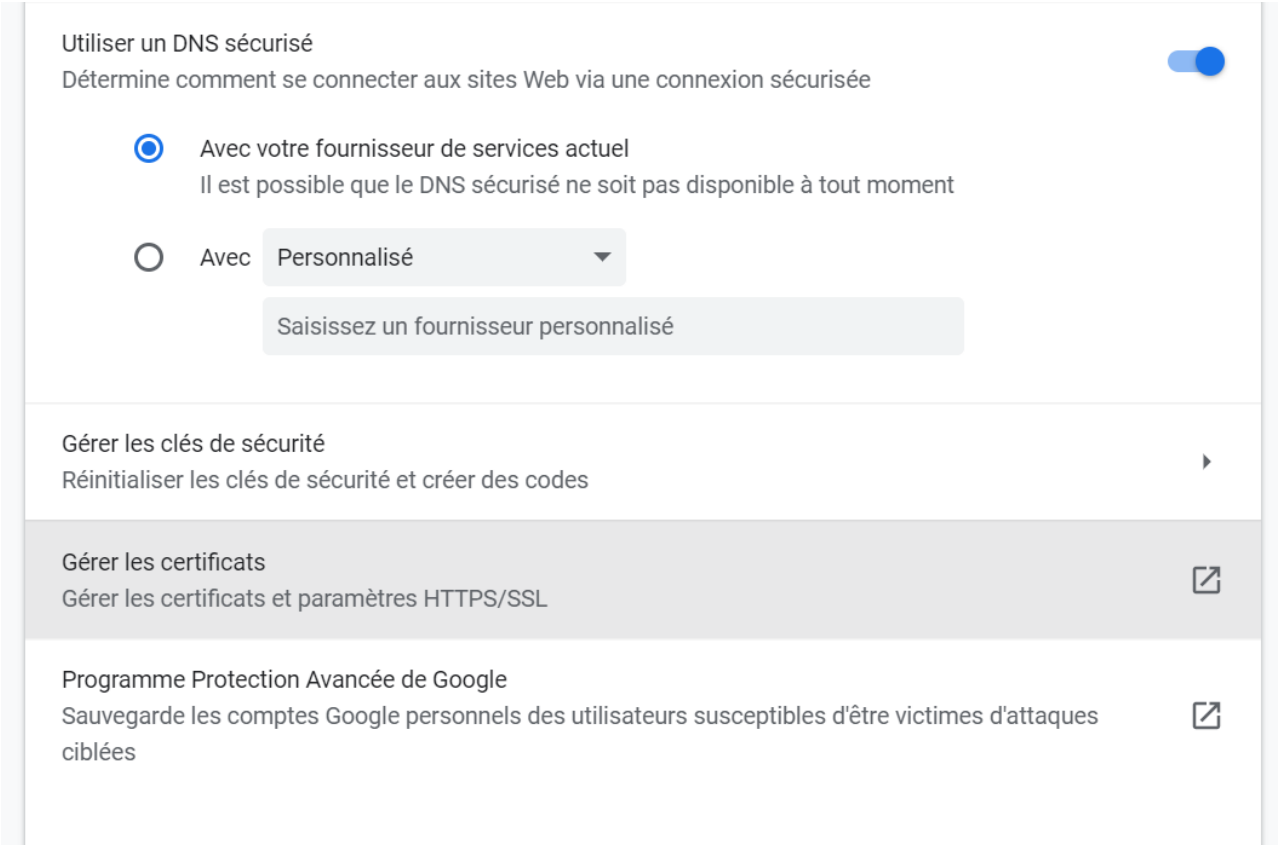
For example for "Google Chrome"

TLS Protocol

Browser

- The browser stores a list of CAs root of trust.

For example for "Google Chrome"

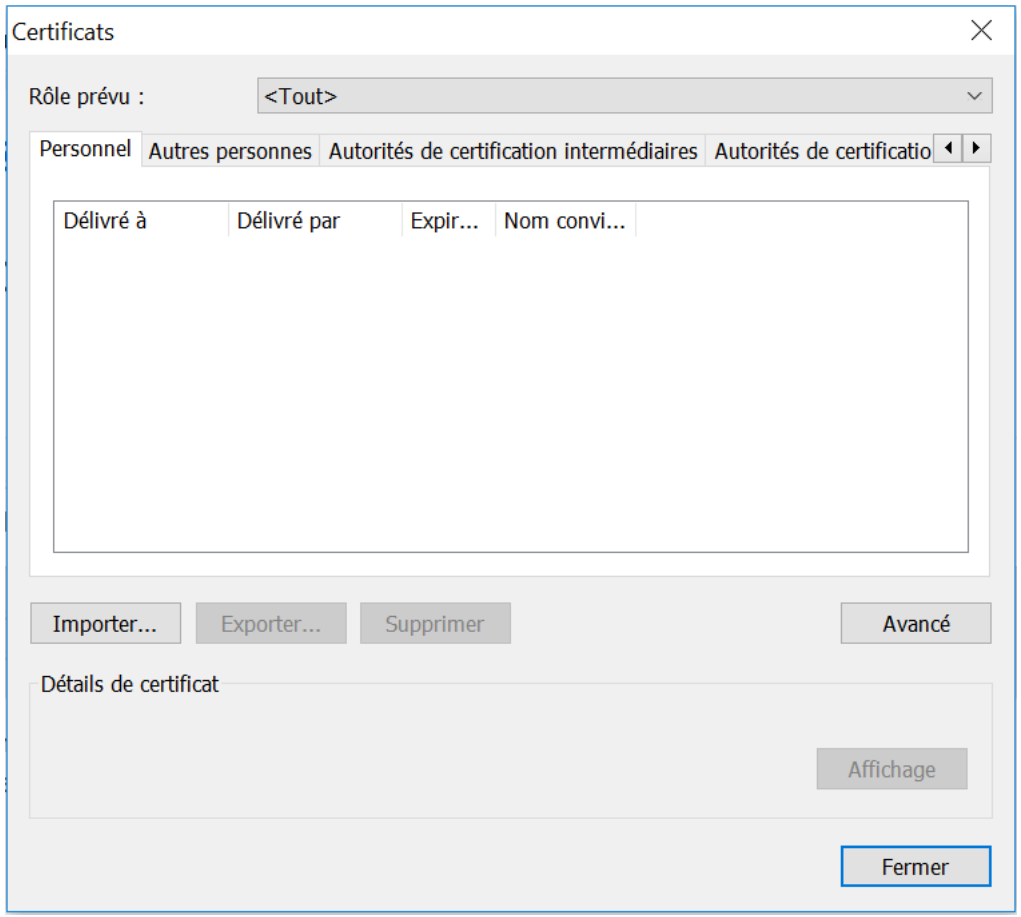


TLS Protocol

Browser

- The browser stores a list of CAs root of trust.

For example for "Google Chrome"

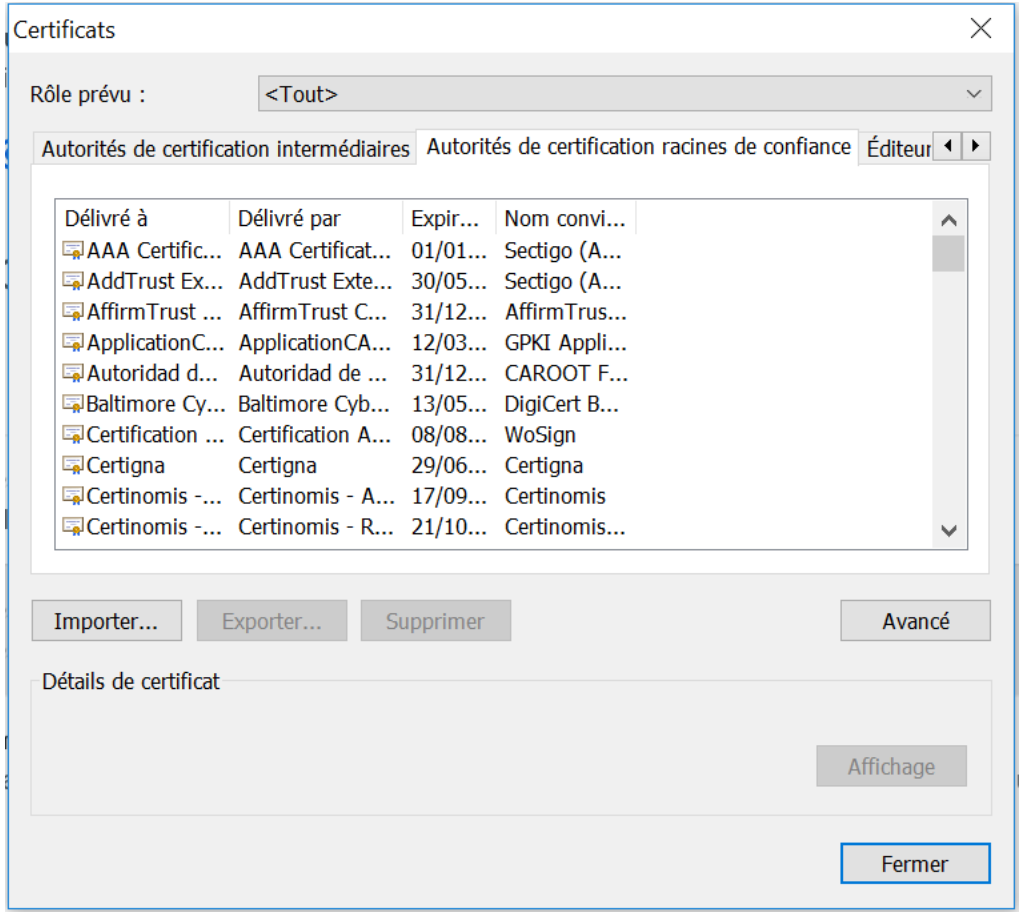


TLS Protocol

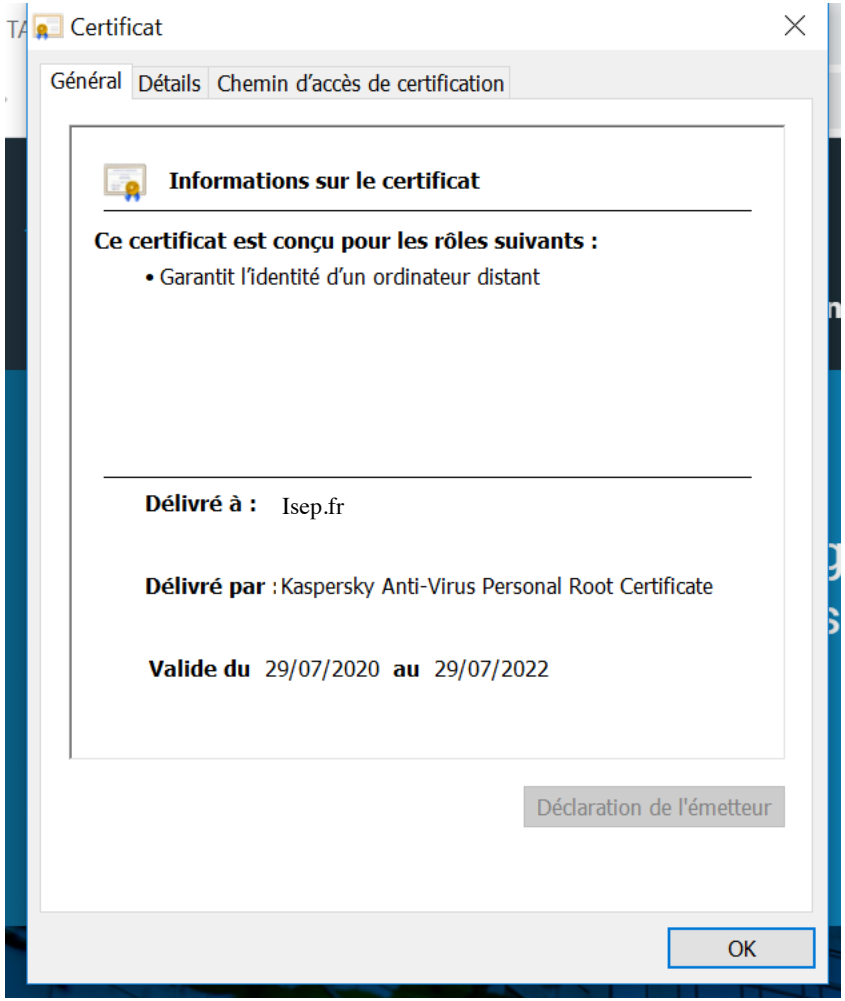
Browser

- The browser stores a list of CAs root of trust.

For example for "Google Chrome"



Example: https://www.isep.fr

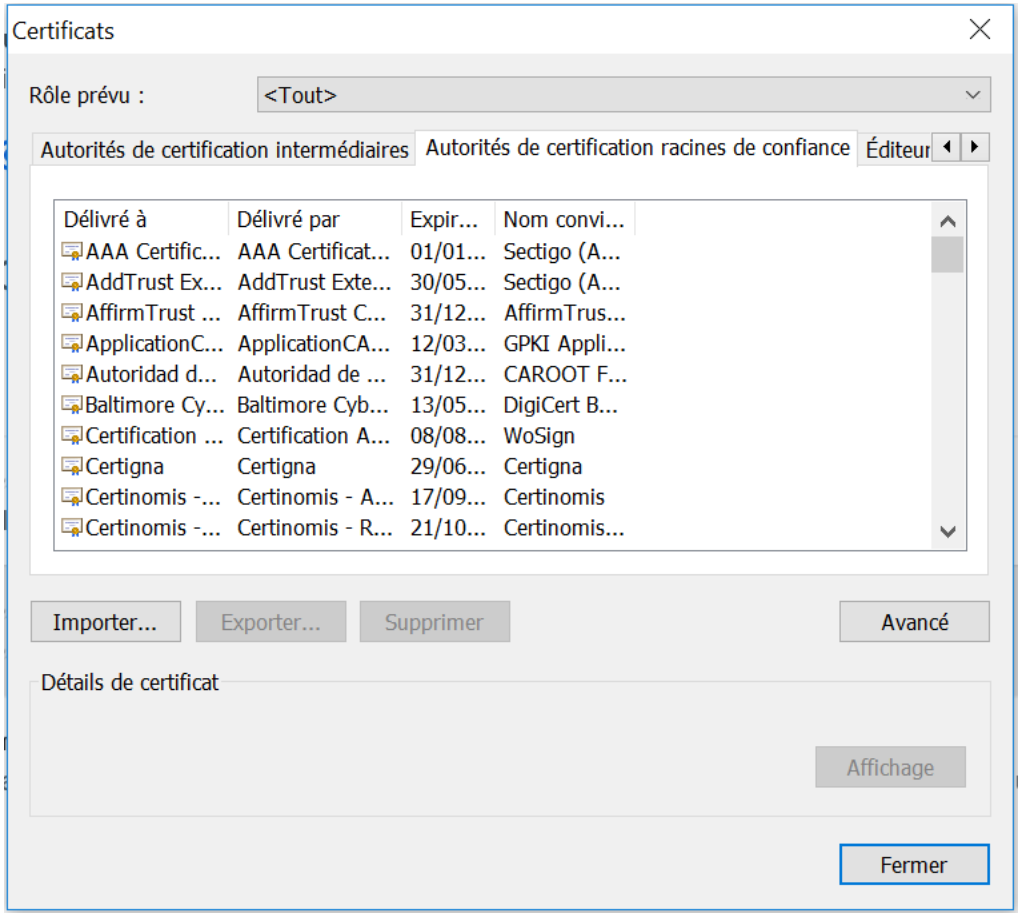


TLS Protocol

Browser

- The browser stores a list of CAs root of trust.

For example for "Google Chrome"

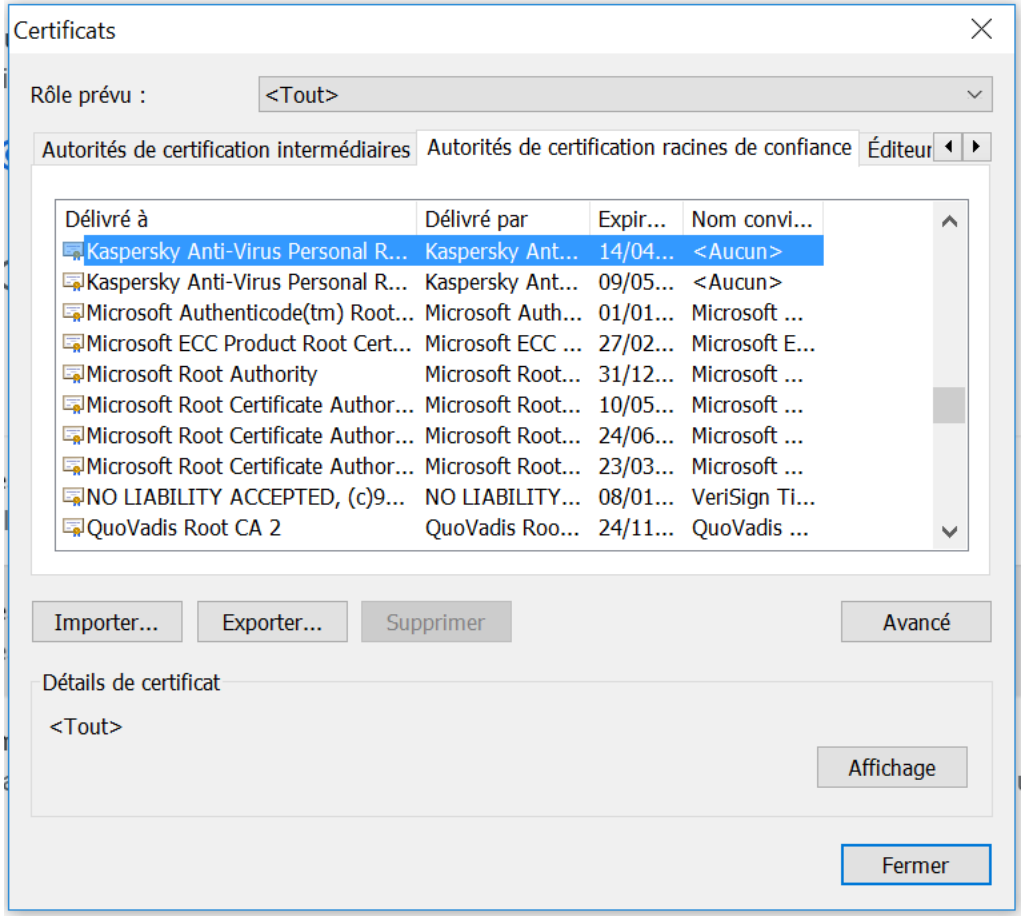


TLS Protocol

Browser

- The browser stores a list of CAs root of trust.

For example for "Google Chrome"

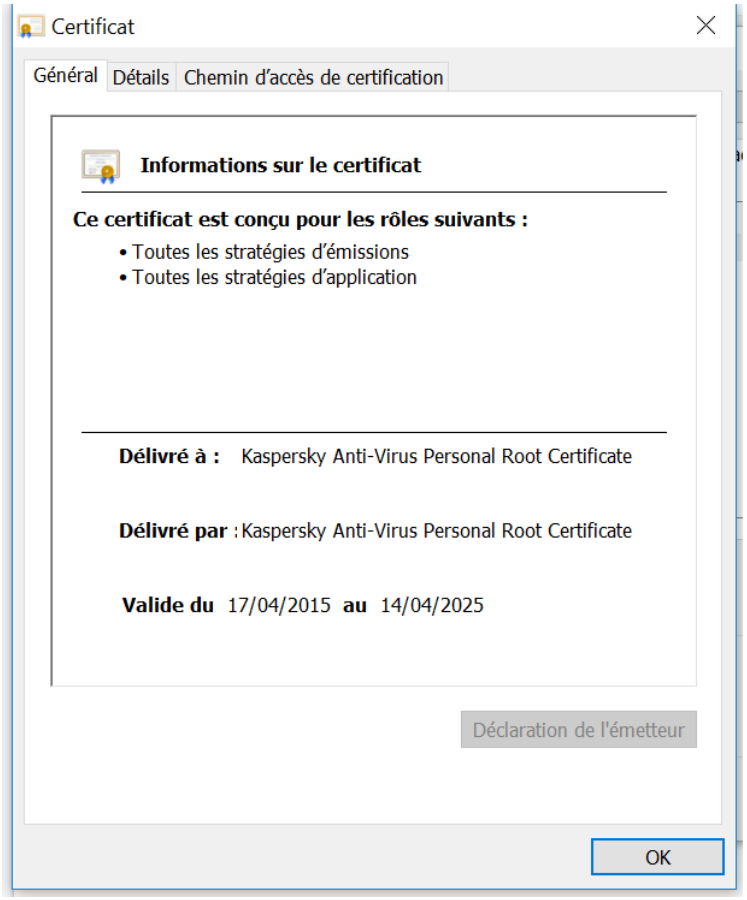


TLS Protocol

Browser

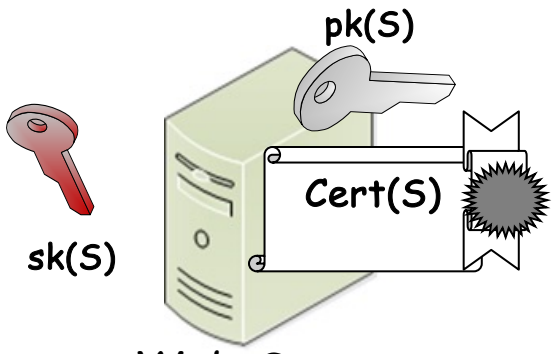
- The browser stores a list of CAs root of trust.

For example for "Google Chrome"



TLS Protocol Session

Browser



Web Server

RNc Generates a Random number RNc

(1) RNc RNc

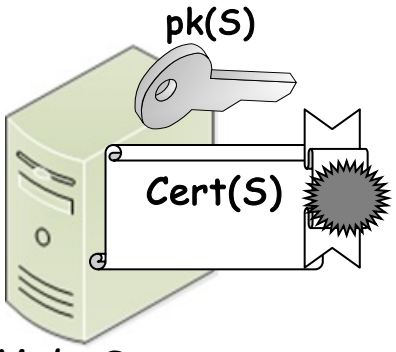
Generates a Random number RNs RNs

(2) RNs RNs

TLS Protocol Session

Browser

sk(S)



Web Server

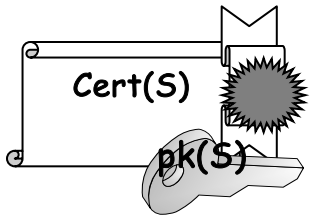
RNs RNc

RNc RNs

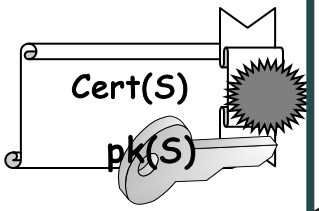
Generates a Signature SignS = {H(RNs, RNc, …)}sk(S)
Using its private key sk(S)

SignS

(3) Server certificate



SingS

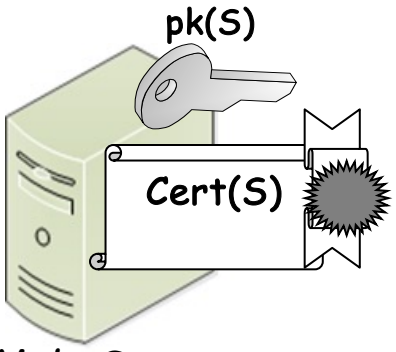


SignS

TLS Protocol Session

Browser

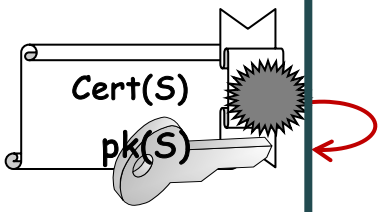
sk(S)



Web Server

RNs RNc

RNc RNs



SignS

Proceeds to authenticate the server as follows:

- Verifies CA is a Trusted Third Party
- Verifies the expiration/revocation of Cert(S)
- Verifies that pk(CA) validates Cert(S)
- Verifies that pk(S) validates SignS

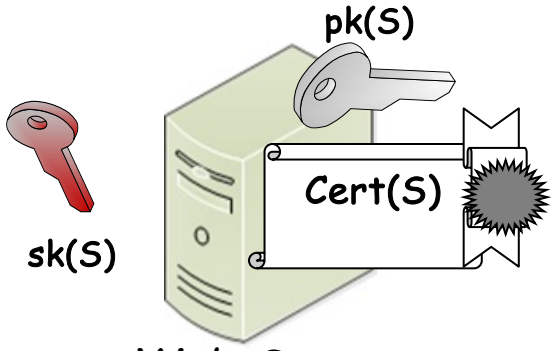
✓ The server is well authenticated

✓ Non-repudiation for the server is well ensured

✓ Integrity of the signed data (RNc, RNs, …) is well ensured

TLS Protocol Session

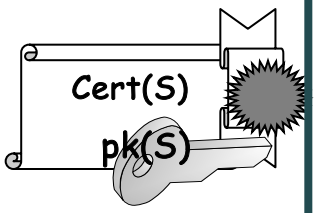
Browser



Web Server

RNs RNc

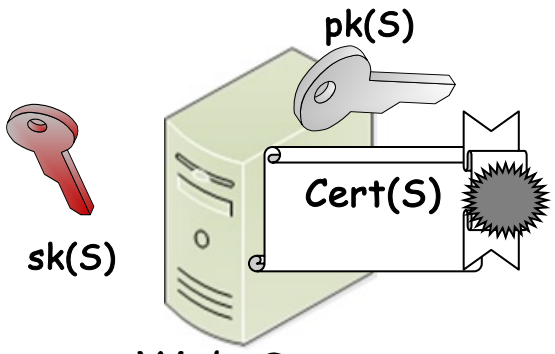
RNc RNs



The server is well authenticated

TLS Protocol Session

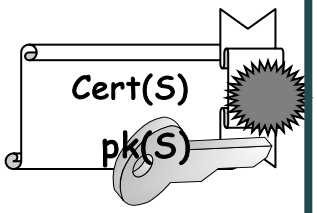
Browser



Web Server

RNs RNc

RNc RNs



The server is well authenticated

PMsc

Generates a Pre-master symmetric secret key PMsc

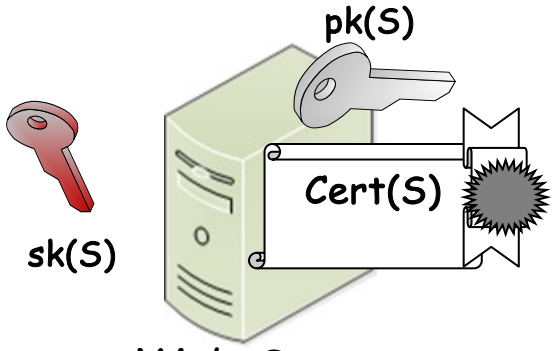
(4) {PMsc} pk(S)

Decrypts using its private key sk(S)

PMsc

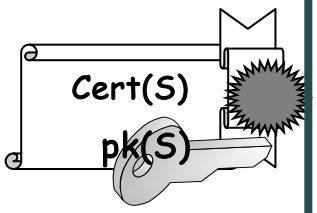
TLS Protocol Session

Browser



Web Server

RNs RNc



The server is well authenticated

PMsc

MS

Both Calculate Master Symmetric Key MS from PMsc, RNc, RNs Using an Hash function

MS = H(PMsc, RNc, RNs)

Objective of MS : is to use a key identified by TLS session while integrating random numbers

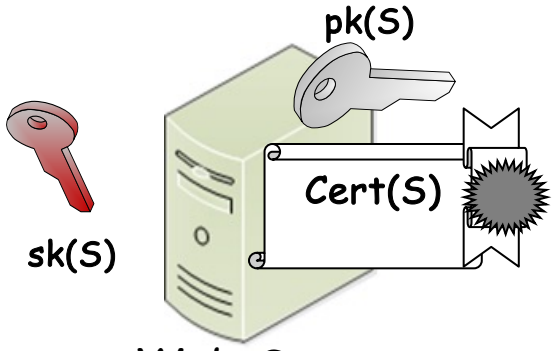
RNc RNs

PMsc

MS

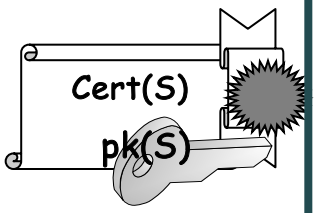
TLS Protocol Session

Browser



Web Server

RNs RNc



The server is well authenticated

PMsc

MS

Both Calculate Master Symmetric Key MS from PMsc, RNc, RNs
Using an Hash function
MS = H(PMsc, RNc, RNs)

(5) {Confidential Data}MS

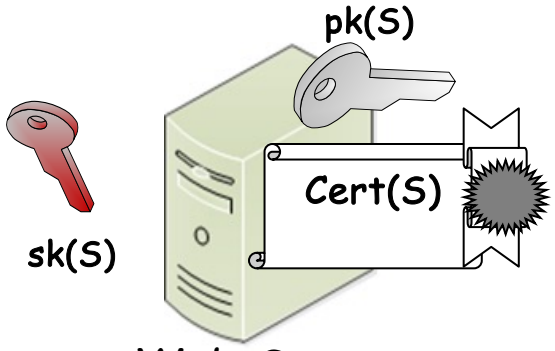
RNc RNs

PMsc

MS

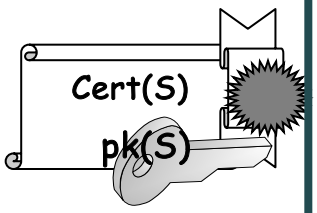
TLS Protocol Session

Browser



Web Server

RNs RNc



The server is well authenticated

PMsc

MS

Both Calculate Master Symmetric Key MS from PMsc, RNc, RNs Using an Hash function

MS = H(PMsc, RNc, RNs)

(6) {Other Confidential Data}MS

RNc RNs

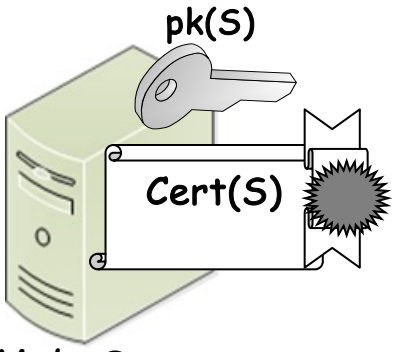
PMsc

MS

TLS Protocol Session

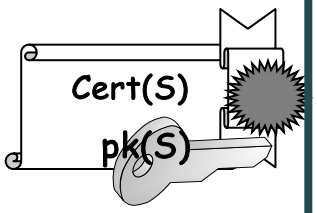
Browser

sk(S)



Web Server

RNs RNc



The server is well authenticated

PMsc

MS

Both Calculate Master Symmetric Key MS from PMsc, RNc, RNs Using an Hash function

MS = H(PMsc, RNc, RNs)

Secure TLS session Thanks to MS (TLS session Key)

RNc RNs

PMsc

MS

Security of the Communication Medium

-Vulnerabilities in Card Payment System-

Vulnerabilities in Card Payment System

Banks offer us the possibility to open a bank account

- Guarantee the fluidity of our financial transactions
- Safely manage our funds
- Use Payment methods such as: bank cards, checks, etc.

Merchants

- *A merchant can open a merchant's account to accept payments*
- *From clients which use their bank payment methods*

Vulnerabilities in Card Payment System

According to [1] [2] [3] [4]:

- The *Bank Card* is confirmed the *fastest and most convenient* payment method in *France*
- It's use has been *steadily increasing since 2000* with *more than 8%* on average *per year*
- In 2015, *71% of French people* say that *Bank Cards* are their *preferred means of payment*
- In 2016, *92% of French people declared* that they *use the Bank Card as their first priority*

[1] Etude de l'institut de sondages d'opinion CSA, "Les français et les moyens de paiement," https://www.economie.gouv.fr/files/sondagecsa_synthese.pdf
[2] Fédération bancaire française, "Les moyens de paiement," http://www.fbf.fr/fr/files/AC3CBC/Les%20Moyens%20de%20Paiement.pdf
[3] La finance pour tous, "La carte bancaire," http://www.lafinancepourtous.com/Banque-auquotidien/Moyens-de-paiement/La-carte-bancaire/
[4] Delphine Cuny, "Carte, virement, chèque ou cash : comment paie-t-on en europe ?" https://www.latribune.fr/entreprises-finance/banques-finance/carte-virement-cheque-ou-cash-comment-paie-t-on-en-europe-750879.html

Vulnerabilities in Card Payment System

Bank card has *much utility* over *cash* and *other payment methods*:

- *Simple* to obtain by banks
- Includes *insurance/assistance*
- *Small in size* and *easy to carry*
- *TO BE PROTECTED:* we can *keep it* in a *safe place*

Vulnerabilities in Card Payment System

Critical Banking Data
- *Name: Nour*
- *PAN (Primary Account Number): 1234 5678….*
- *Expiration date: 12/2023*
- *Security code: 333*

Bank card has *much utility* over *cash* and *other payment methods*:

- It *stores critical banking data* that *are needed primarily* to perform:

Online Payment

Contact payment

Contactless-NFC payment (5 cm)

Magnetic Stripe Payment

PoS: Point of Sale

Vulnerabilities in Card Payment System

- ***Bank card*** is the *Fastest, preferred, first priority* and *most convenient* payment method in *France*

- ***Bank card*** has *much utility* over *cash* and *other payment methods*

Clients *consider* that the ***Bank Card*** as a *crucial* and *magical solution*

to safely *manage their funds* and to *protect themselves*

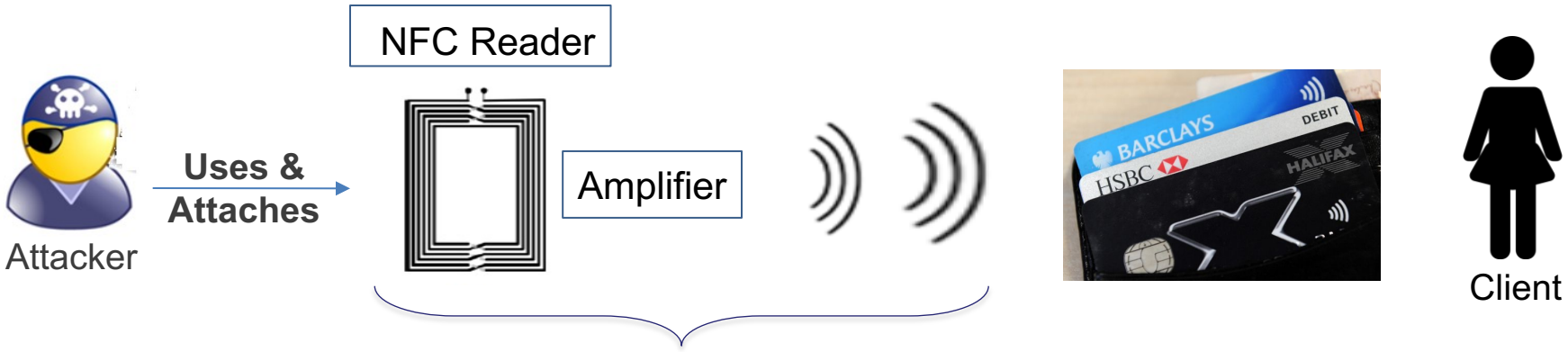
Vulnerabilities in Card Payment System

✓ Clients *assume* that this *magic card* is *very secure/reliable* because:

- It *stores* the *critical banking data* in a *smart chip*
- Smart-Chip offers a *highly secure environment*

Vulnerabilities in Card Payment System

Authors in the studies [5] [6] [7] [8] [9] prove that the *assumption of clients is not completely accurate* by *demonstrating the following attack*:



✓ Reach a *distance of NFC reading up to 1.50 meters*

✓ Steal the ***Banking Data (PAN & ExpDate)*** remotely:

- without stealing the physical bank card
- without the knowledge of the cardholder

NFC Attack

[5] M. Emms and A. van Moorsel, "Practical attack on contactless payment cards," HCI2011 Workshop Heath, Wealth and Identity Theft, 2011.
[6] B. Cohen, "Millions of barclays card users exposed to fraud," https://www.channel4.com/news/millions-of-barclays-card-users-exposed-to-fraud , 2012.
[7] R. Lifchitz, "Hacking the nfc credit cards for fun and debit," Hackito Ergo Sum conference, April 2012.
[8] Gerard Tubb, "Contactless cards: App reveals security risk," https://news.sky.com/story/contactlesscards-app-reveals-security-risk-10443980 , 2013.
[9] M. J. Emms, "Contactless payments: usability at the cost of security?," Ph.D.Thesis, Newcastle University, 2016.

Vulnerabilities in Card Payment System

NFC Attack 

This kind of attack *has raised our attention* to ask *two important questions*:

(1) *How this attack can be produced ?*

(2) *How can the malicious adversary exploit the stolen banking data ?*

"In order to address these questions"

*How the **Bank Card** can communicate with an **NFC reader** ?*

Vulnerabilities in Card Payment System

*How the **Bank Card** can communicate with an **NFC reader** ?*

EMV Security Standard

EMV: Europay MasterCard Visa

PoS: Point of Sale

Vulnerabilities in Card Payment System

(1) *How this attack can be produced ?*

~~*Confidentiality of Banking data*~~ **Vulnerability** 1 **Vulnerability** 2 ~~*Authentication of PoS to the Card*~~

~~*Authentication of P to C*~~

NFC Reader

Attacker

Uses & Attaches

Amplifier

~~*Confidentiality of banking data*~~

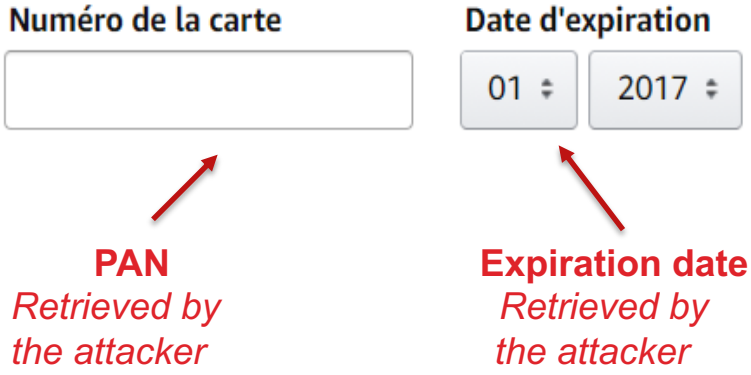
Client

Vulnerability 3 *Because of NFC technology*

Vulnerabilities in Card Payment System

(2) *How can the malicious adversary exploit the stolen banking data ?*

Fraudulent purchase transactions on the internet [10][11][12]



Assumptions:
- *The victim has funds in his bank account*
- *The merchant of the website does not use any additional security mechanism*

*Several websites as **"www.amazon.com"**, **"www.zappos.com"** do not request the security code*

[10] N. El Madhoun and G. Pujolle, “Security enhancements in EMV protocol for NFC mobile payment,” 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom), pp. 1889–1895, 2016
[11] M. Emms and A. van Moorsel, “Practical attack on contactless payment cards,” HCI2011 Workshop Heath, Wealth and Identity Theft, 2011
[12] R. Lifchitz, “Hacking the nfc credit cards for fun and debit,” Hackito Ergo Sum conference, April 2012

Vulnerabilities in Card Payment System

(2) *How can the malicious adversary exploit the stolen banking data ?*

Attack by brute force [13]

Attacker

PAN *4387 5689 9123 4567* → ***Brute force*** → ***Expiration date***

Brute force → ***Security Code***

- Using a website:
  - that asks for *the security code*
  - *does not block* this type of brute force attack

- All possibilities one thousand: *000 to 999*
- This operation can be relatively fast *if it is not blocked by servers*

[13] M. A. Ali, B. Arief, M. Emms, and A. van Moorsel, “Does the online card payment landscape unwittingly facilitate fraud?,” IEEE Security & Privacy, pp. 78–86, 2017.

Vulnerabilities in Card Payment System

(2) *How can the malicious adversary exploit the stolen banking data ?*

Attack by stealing merchant's proofs

Client's Proof

Merchant's Proof

Vulnerabilities in Card Payment System

(2) *How can the malicious adversary exploit the stolen banking data ?*

Attack by stealing merchant's proofs

A thief can **easily steal the merchant's receipts** and obtain the **banking data of several clients** insofar **as the merchants**:

- Generally **do not protect** these proofs **conscientiously**
- Will not need **these proofs anymore, every 12-13 months,** and they will be **able to throw them away**

Security of the Communication Medium

- EMV Security Protocol -

EMV Security Protocol

EMV Security Standard

EMV: Europay MasterCard Visa

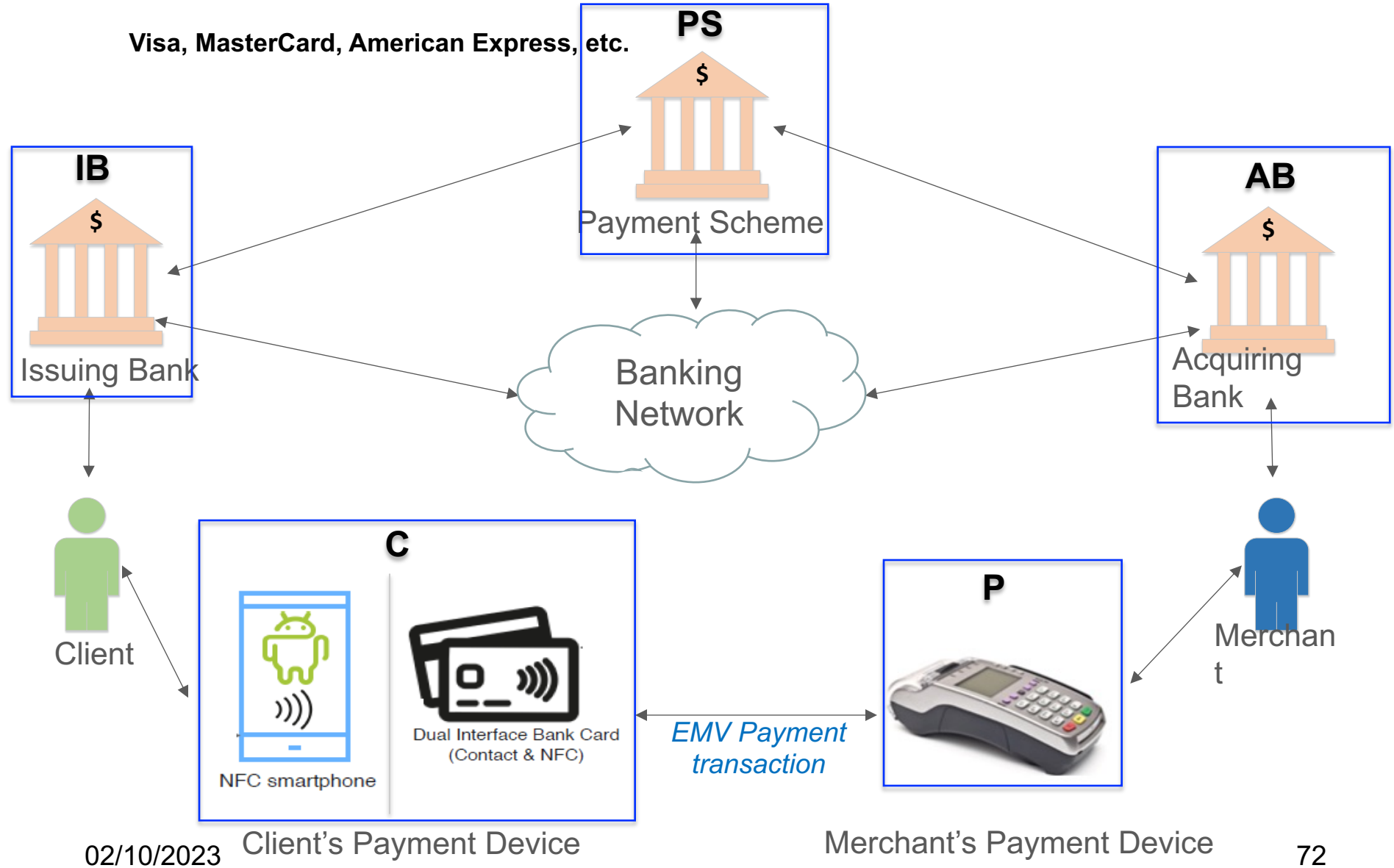
EMV Security Protocol

EMV is the protocol that allows to secure the communication during a purchase transaction (with contact or contactless-NFC) between:

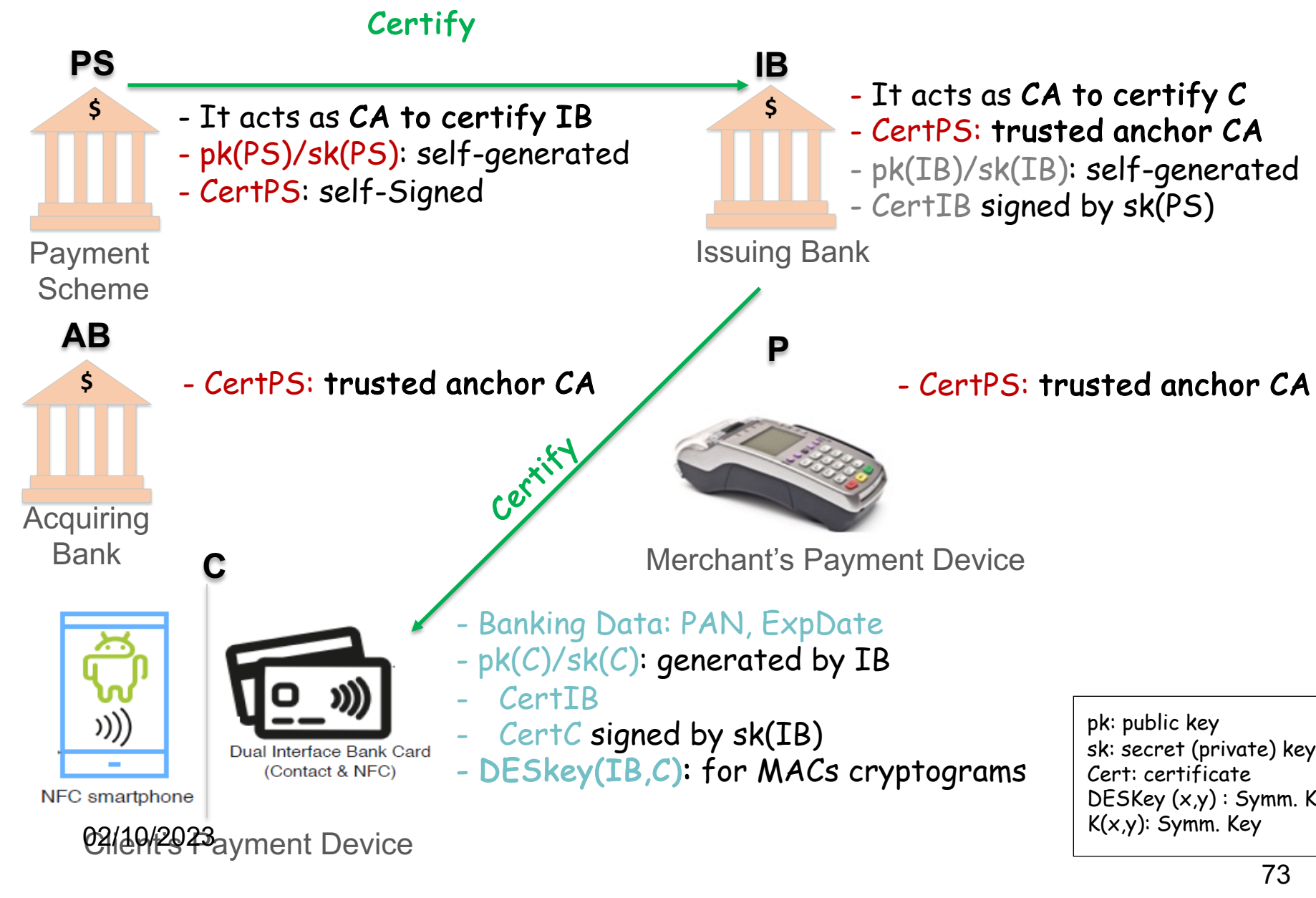
- A Client's Payment Device: Bank card or an NFC smartphone (emulating a bank card)

- A Merchant's Payment Device: Point of Sale (PoS)

EMV Actors

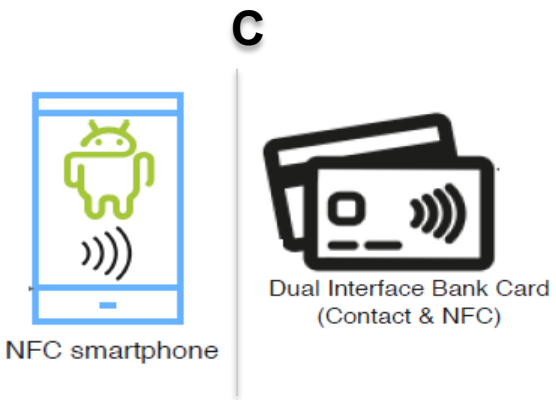

Visa, MasterCard, American Express, etc.
PS
$
Payment Scheme
IB
$
Issuing Bank
AB
$
Acquiring Bank
Banking Network
Client
C
NFC smartphone
Dual Interface Bank Card (Contact & NFC)
EMV Payment transaction
P
Merchant
Client's Payment Device
Merchant's Payment Device

EMV Security Elements



pk: public key
sk: secret (private) key
Cert: certificate
DESKey (x,y) : Symm. Key
K(x,y): Symm. Key

EMV Security Elements



Client's Payment Device

- DESkey(IB,C): for MACs cryptograms → **Used in the EMV payment transaction**

pk: public key
sk: secret key
Cert: certificate
DESKey: Symm. Key

EMV Protocol Session

- In order to perform a **secure EMV transaction: Contact or Contactless-NFC,**
- **EMV actors** exchange **security messages** that can be **divided into 4 steps**:

EMV Phase 1: Initialization

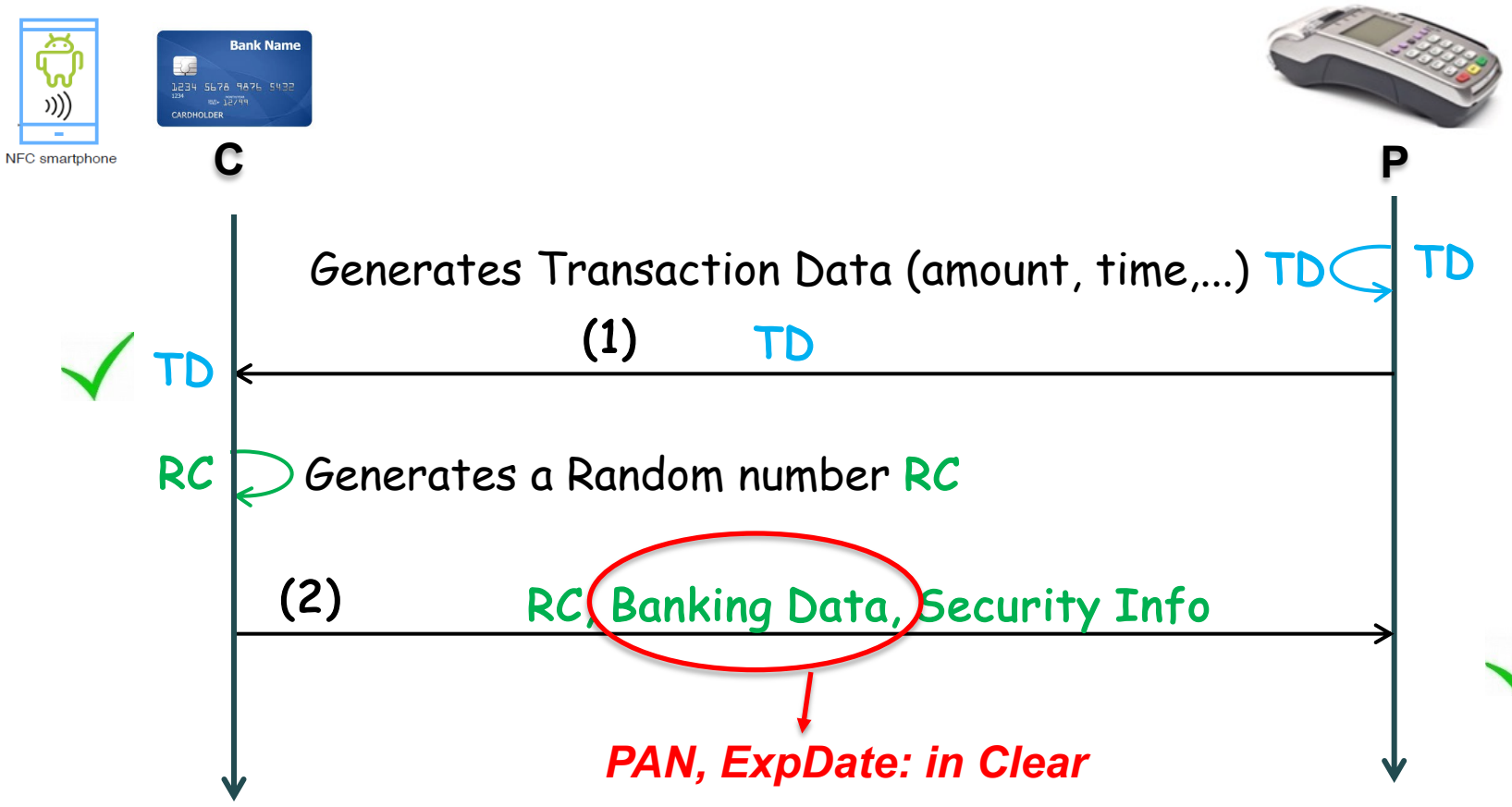
EMV Phase 2: Authentication of C to P

EMV Phase 3: Authentication of the client (User)

EMV Phase 4: Actual Transaction (Online/Offline)

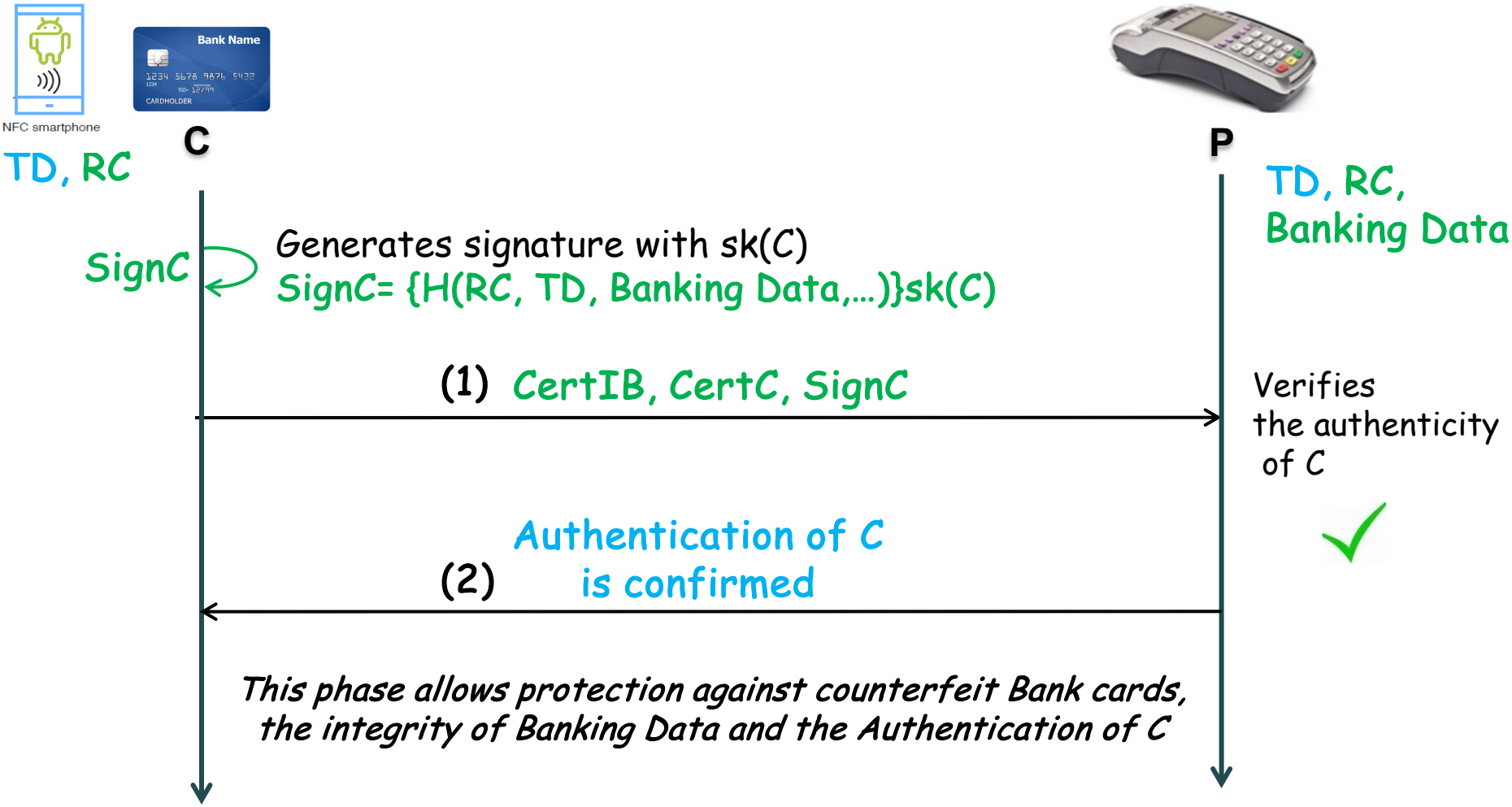
EMV Protocol Session

EMV Phase 1: Initialization



EMV Protocol Session

EMV Phase 2: Authentication of C to P



EMV Protocol Session

EMV Phase 2: Authentication of C to P

Exercise 1:

How P verifies the authenticity of C (what are the steps) ?

EMV Protocol Session

EMV Phase 3: Authentication of the client (User)

By a PIN or a Hand Signature

NFC smartphone

C

Client

Enters a PIN code or signature to authenticate itself to P

This phase allows protection against lost and stolen bank cards
And the authentication of the client (user)

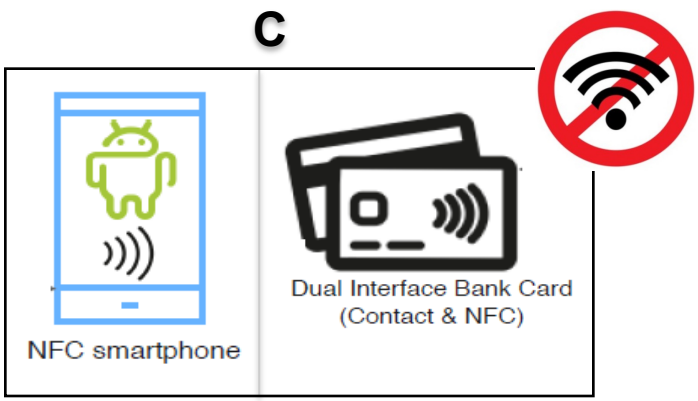
P

EMV Protocol Session

EMV Phase 4: Actual Transaction (Online/Offline)

This phase ensures to P that the transaction is confirmed and authorized by IB

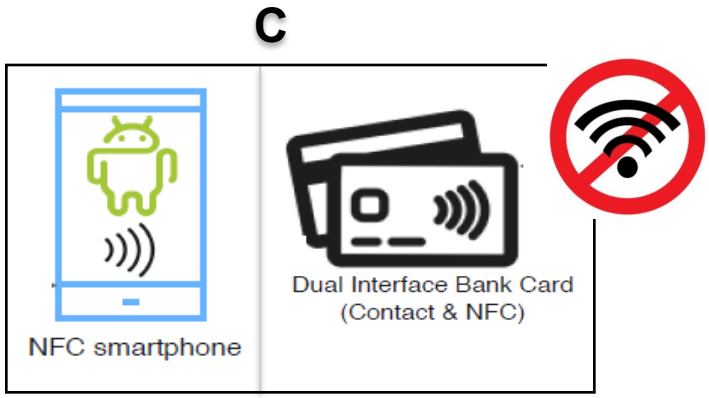
Case 1



Client's Payment Device

Merchant's Payment Device

Case 2

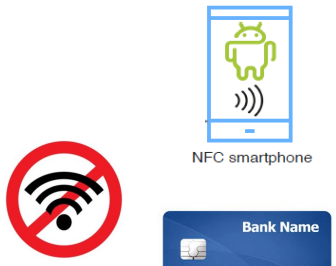


Client's Payment Device

Merchant's Payment Device

EMV Protocol Session

EMV Phase 4: Actual Transaction (Online/Offline)



P is Online

C

P

TD, RC

TD, RC,
Banking Data

MACKey Generates MACKey = H(DESKey(IB,C), PAN, RC)

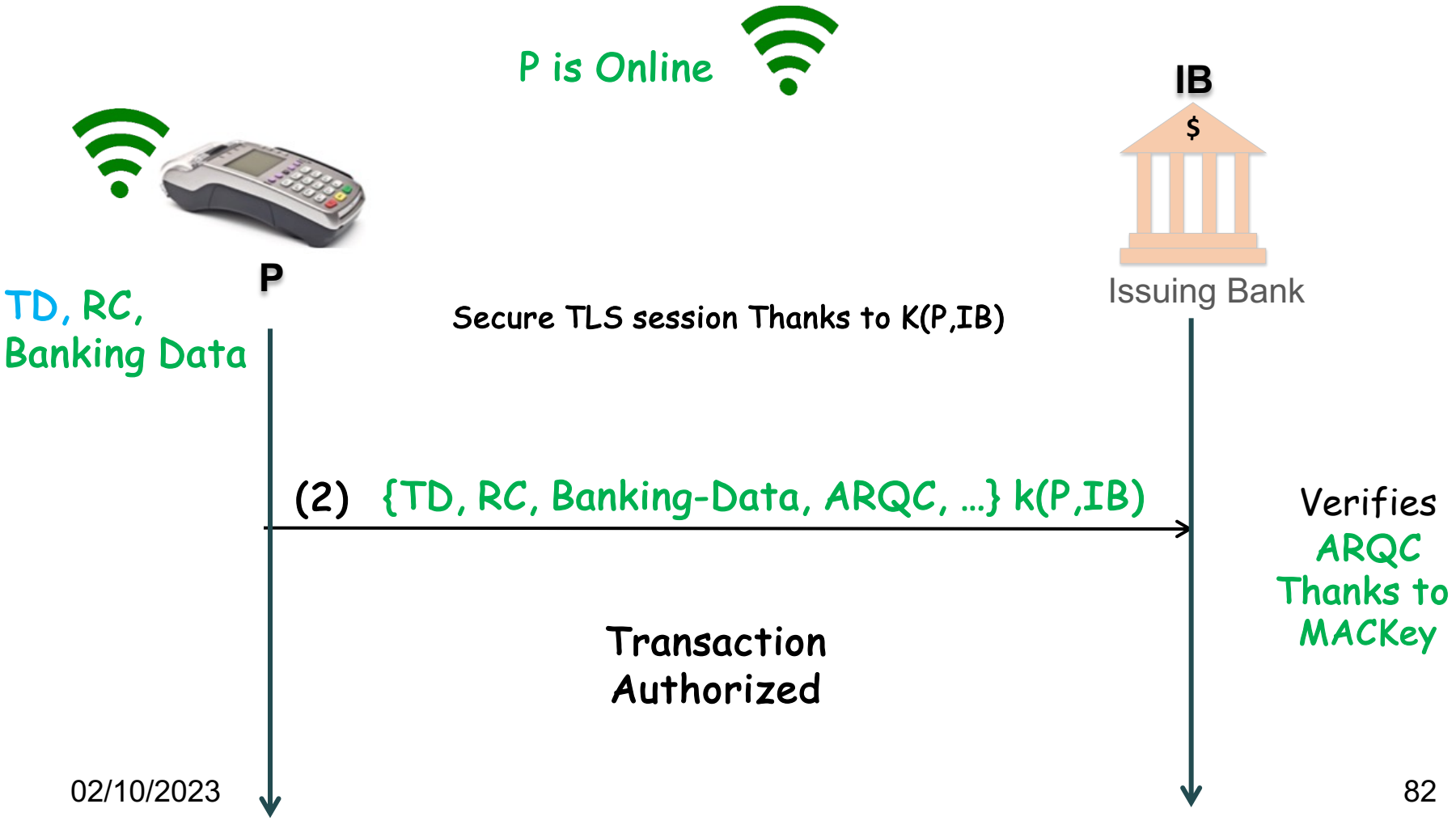
ARQC Generates
ARQC = {H(RC, TD, Banking Data, ...)} MACKey

(1) ARQC

Cannot Verify
ARQC
(does not have
DESKey(IB,C))

EMV Protocol Session

EMV Phase 4: Actual Transaction (Online/Offline)



EMV Protocol Session

EMV Phase 4: Actual Transaction (Online/Offline)

Exercise 2:

How IB verifies ARQC to authorize the transaction (what are the steps) ?

Thanks !

Nour EL MADHOUN

Associate Professor

nour.el-madhoun@isep.fr