Chapter 4

PKI and Key Management

Objectives

- □Differentiate the different key types
- □Identify the key management lifecycle
- □Understand the importance of digital certificates
- □Understand the role of PKI
- □Identify PKI architecture and components

Contents

❑ Types of cryptographic keys

❑ Robustness of cryptographic keys

❑ Digital Certificates

❑ PKI components

❑ PKI architectures

Introduction

❑ A cryptosystem remains secure as well as:

- The cryptographic keys is protected (confidential).
- The cryptographic algorithm is robust against cryptanalysis.
- The key is unpredictable.

A cryptographic key should be secured all along his lifecycle:

- Generation
- Storage
- Distribution
- destruction

Key Management

❑ Poor key management may easily lead to compromise of systems where the security is based on cryptography.

Cryptographic keys

Several different types of keys are defined. The keys are identified according to:

1. their classification as public, private or symmetric keys,
2. their use.
3. Static (long life) or ephemeral (short life)

❑19 different types of cryptographic keys defined in: NIST Special Publication 800-57, Part 1, “Recommendation for Key Management”

Key Usage

- a single key **shall be used for only one purpose** **(e.g., encryption, integrity authentication, key wrapping, random bit generation, or digital signatures). There are several reasons for this:**

  1. The use of the same key for two different cryptographic processes may weaken the security provided by one or both of the processes.
  2. Limiting the use of a key limits the damage that could be done if the key is compromised.
  3. Some uses of keys interfere with each other. e.g. an asymmetric key pair should only be used for either encryption or digital signatures, not both.

Crypto period

- The crypto period is the time spent during which a specific key is authorized for use.

- The crypto period is important because it:
  - Limits the amount of information, protected by a given key, that is available for cryptanalysis.
  - Limits the amount of exposure and damage, should a single key be compromised.
  - Limits the use of a particular algorithm to its estimated effective lifetime.

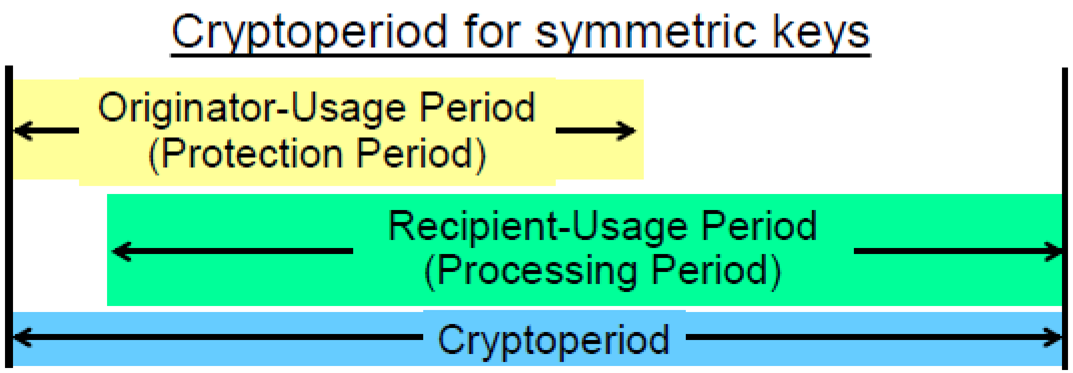
Crypto period trade-off

❑ In general, as the sensitivity of the information or the criticality of the processes increases, the crypto-period should decrease in order to limit the damage resulting from compromise.

❑ Short crypto-periods may be counter-productive, particularly where denial of service is the paramount concern, and there is a significant overhead and potential for error in the re-keying, key update or key derivation process.

❑The crypto-period is therefore a **trade-off**

Key Usage Periods


Cryptoperiod for symmetric keys
Originator-Usage Period (Protection Period)
Recipient-Usage Period (Processing Period)
Cryptoperiod

❑ A key can be used for protection and/or processing.

- ➢ Protection: Key is e.g. used to encrypt or to generate digital signature
- ➢ Processing: Key is e.g. used to decrypt or to validate digital signature

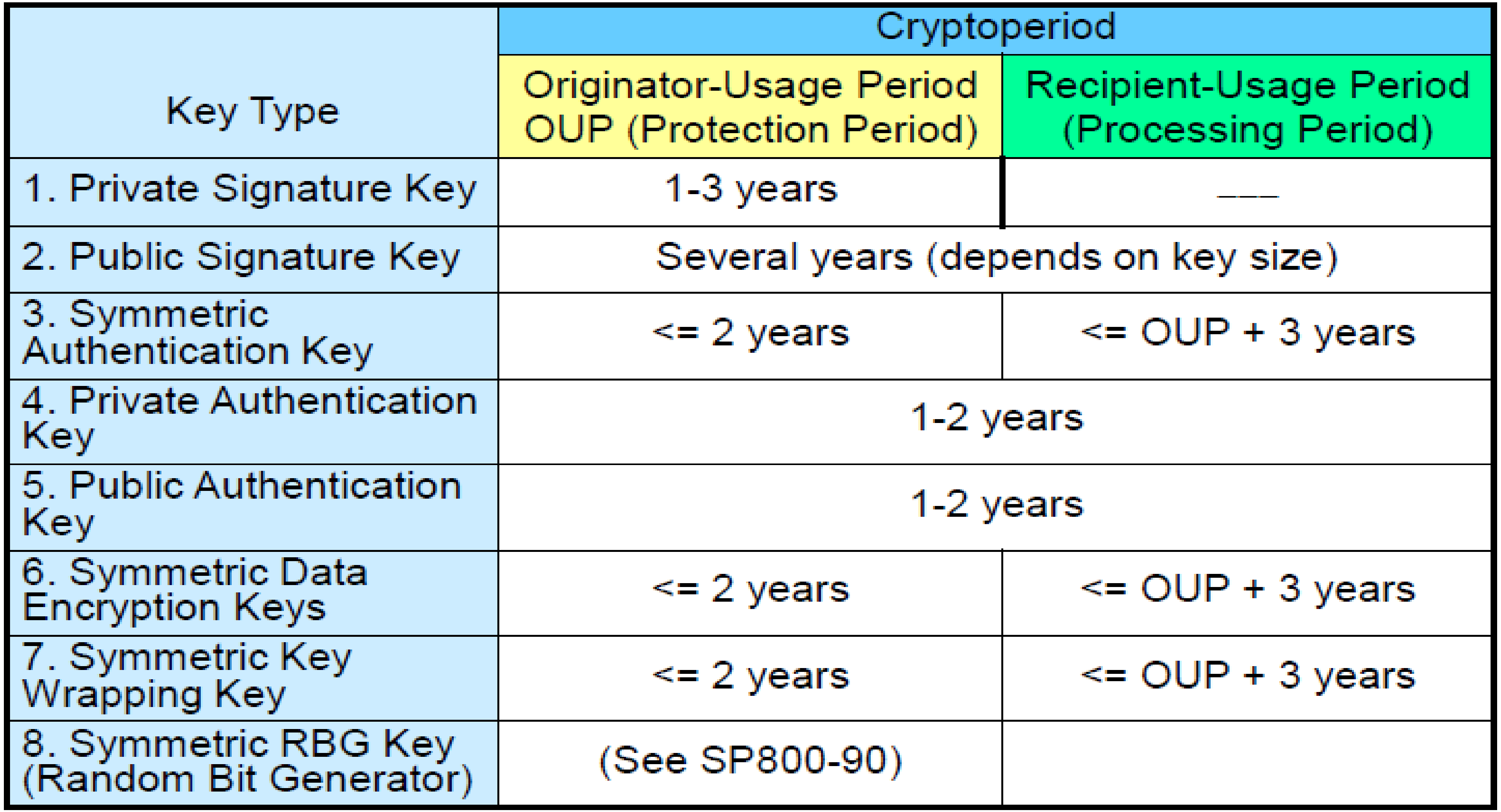
❑ The **crypto-period lasts from the beginning of the protection period to the end of the processing period.**

❑ A key **shall not be used outside of its specified period.**

❑ The processing period can continue after the protection period.

Recommended crypto periods

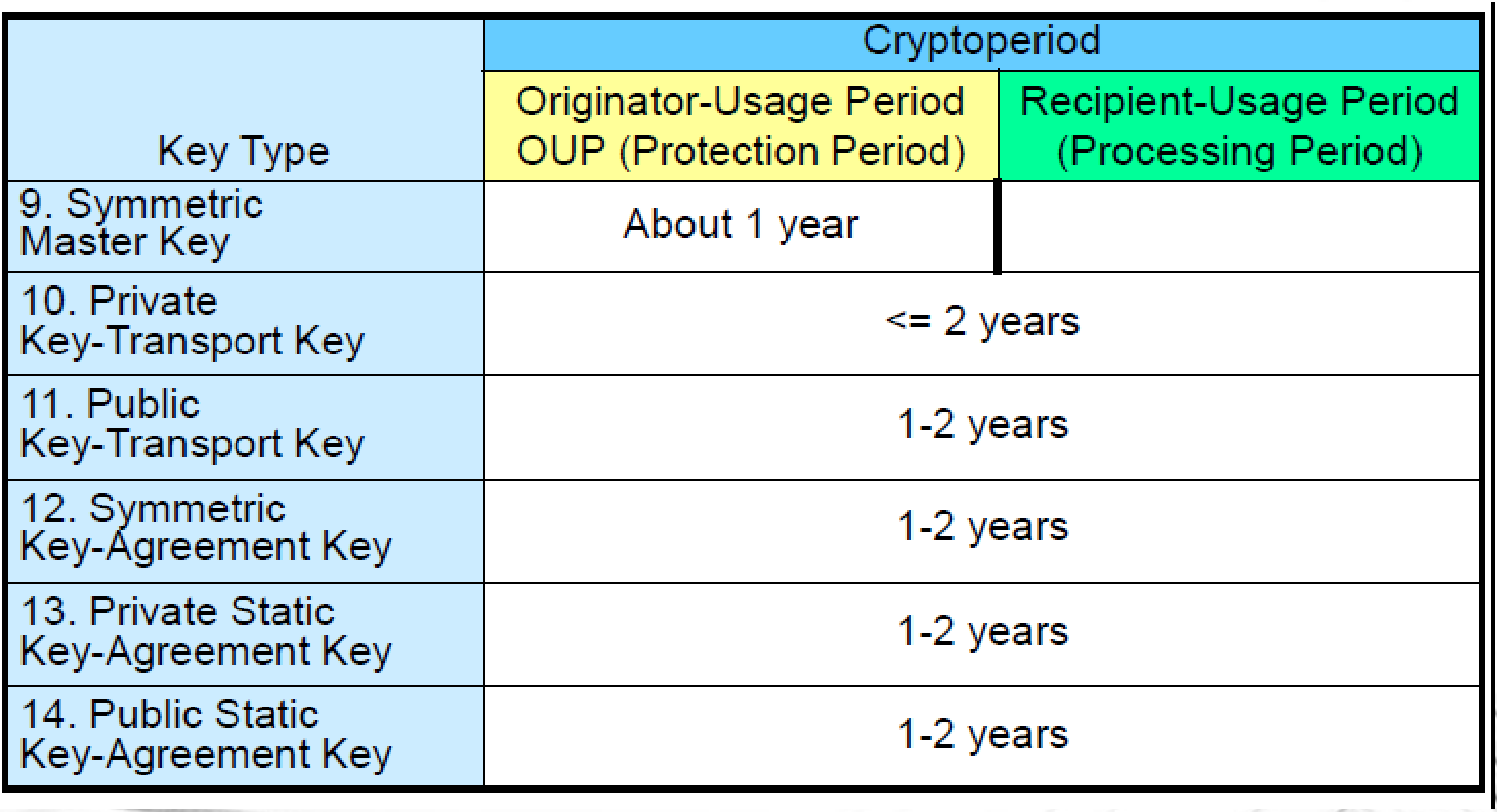
(NIST SP 800-57)



Key Type	Cryptoperiod	
	Originator-Usage Period OUP (Protection Period)	Recipient-Usage Period (Processing Period)
1. Private Signature Key	1-3 years	———
2. Public Signature Key	Several years (depends on key size)	
3. Symmetric Authentication Key	<= 2 years	<= OUP + 3 years
4. Private Authentication Key	1-2 years	
5. Public Authentication Key	1-2 years	
6. Symmetric Data Encryption Keys	<= 2 years	<= OUP + 3 years
7. Symmetric Key Wrapping Key	<= 2 years	<= OUP + 3 years
8. Symmetric RBG Key (Random Bit Generator)	(See SP800-90)	

Recommended crypto periods

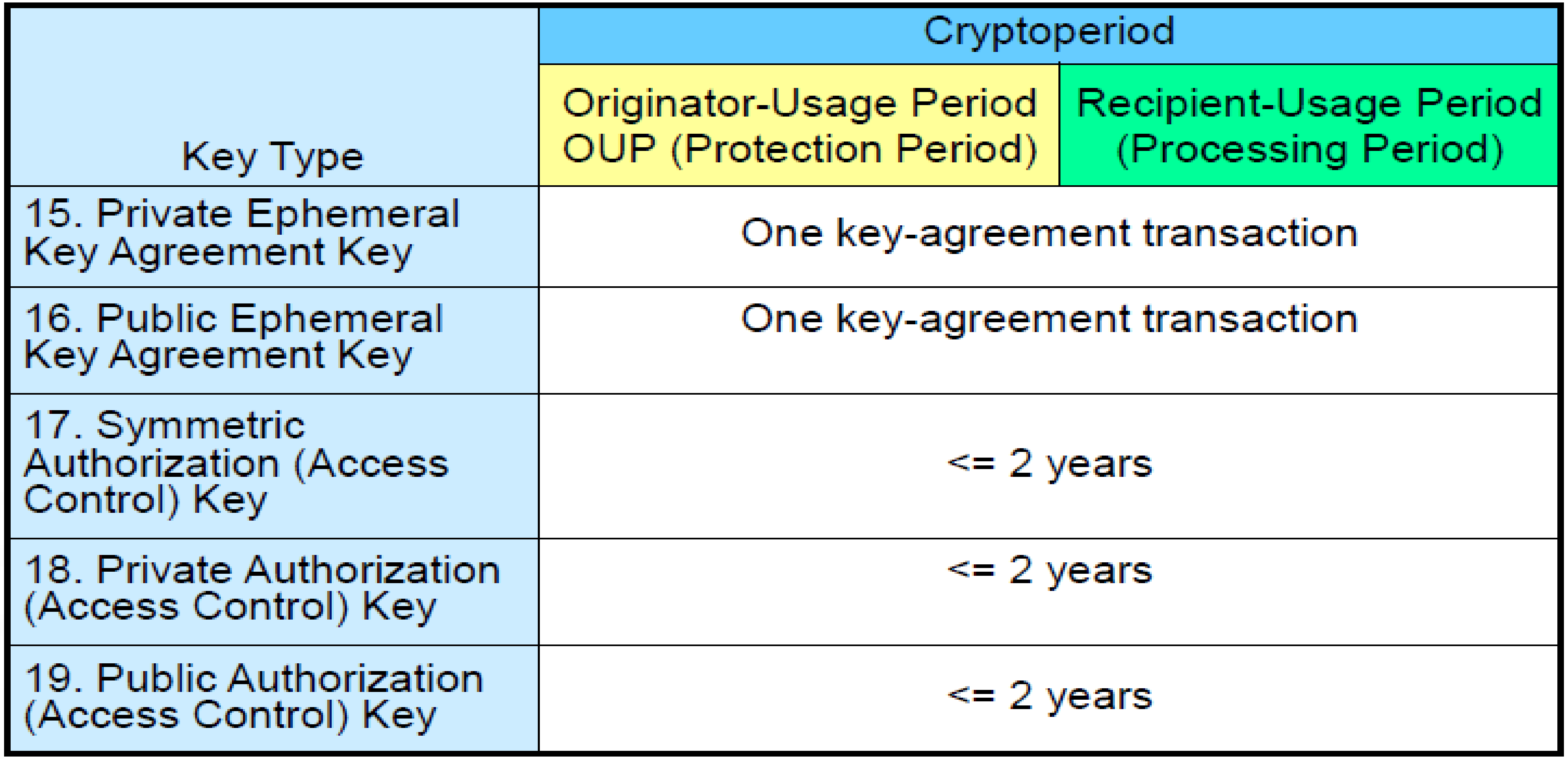
(NIST SP 800-57)



Key Type	Cryptoperiod	
	Originator-Usage Period OUP (Protection Period)	Recipient-Usage Period (Processing Period)
9. Symmetric Master Key	About 1 year	
10. Private Key-Transport Key	<= 2 years	
11. Public Key-Transport Key	1-2 years	
12. Symmetric Key-Agreement Key	1-2 years	
13. Private Static Key-Agreement Key	1-2 years	
14. Public Static Key-Agreement Key	1-2 years	

Recommended crypto periods

(NIST SP 800-57)



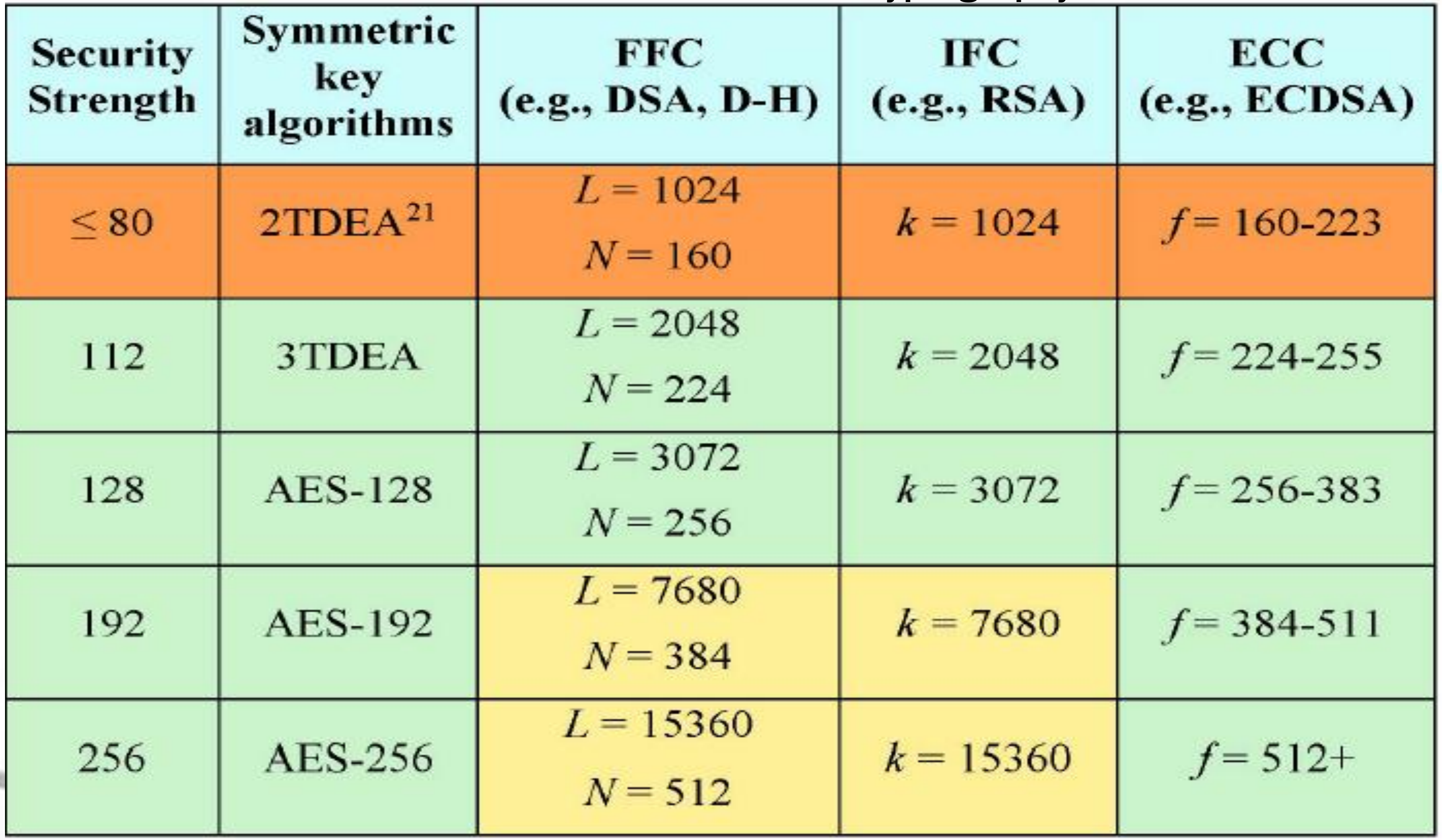
Key Type	Cryptoperiod	
	Originator-Usage Period OUP (Protection Period)	Recipient-Usage Period (Processing Period)
15. Private Ephemeral Key Agreement Key	One key-agreement transaction	
16. Public Ephemeral Key Agreement Key	One key-agreement transaction	
17. Symmetric Authorization (Access Control) Key	<= 2 years	
18. Private Authorization (Access Control) Key	<= 2 years	
19. Public Authorization (Access Control) Key	<= 2 years	

Key Strength comparison

Finite Field Cryptography

Integer Factorization Cryptography

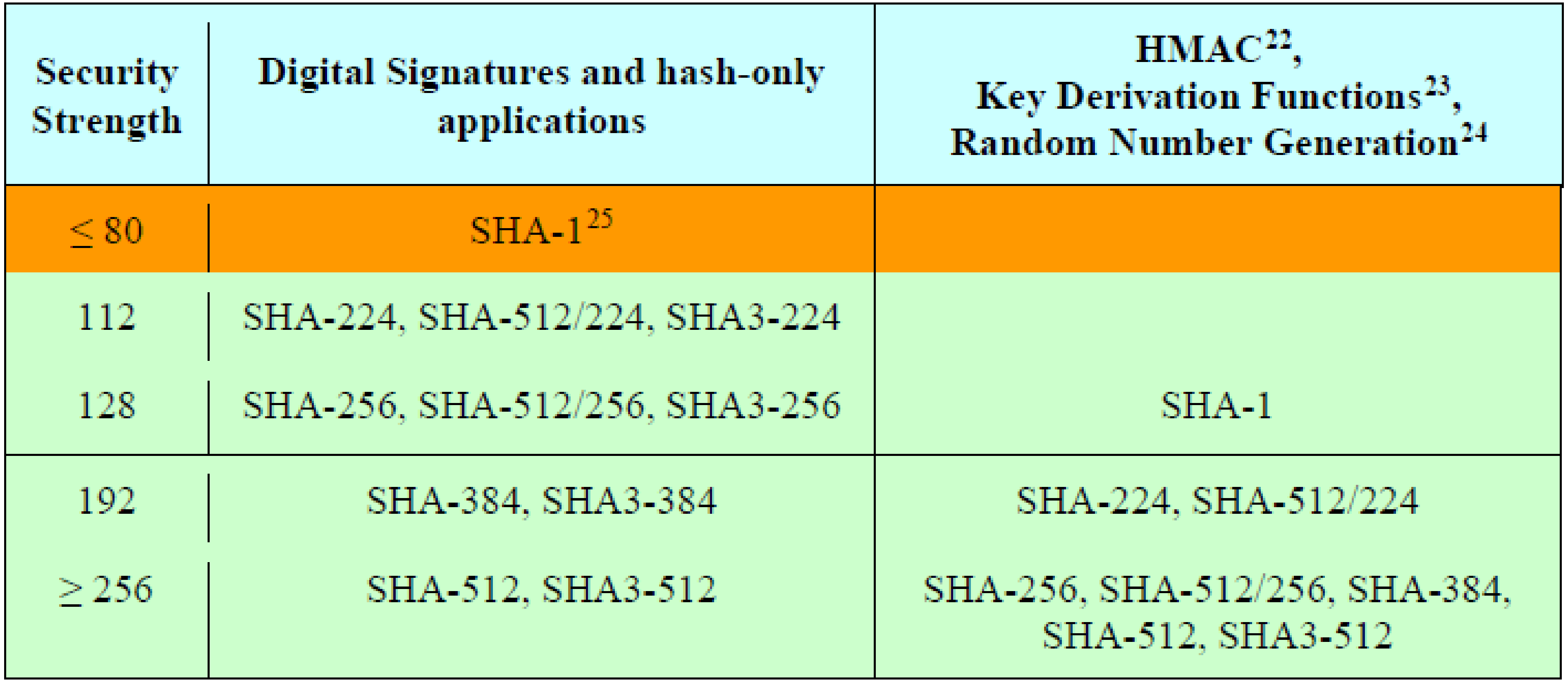
Elliptic Curve Cryptography



Security Strength	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80	2TDEA[21]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

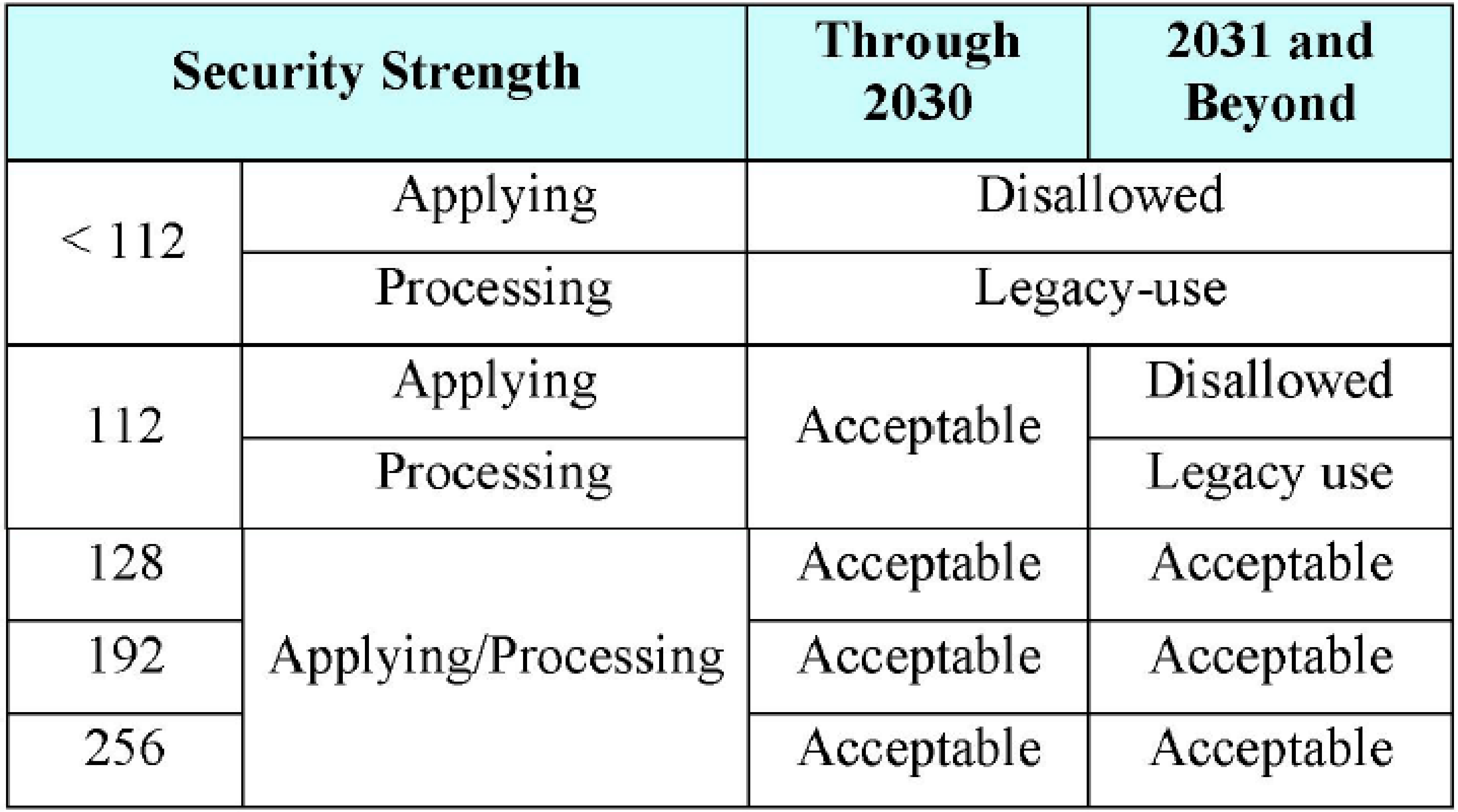
***L* is the size of the public key, *N* is the size of the private key.**

Hash functions strength functions



Security Strength	Digital Signatures and hash-only applications	HMAC[22], Key Derivation Functions[23], Random Number Generation[24]
≤ 80	SHA-1[25]	
112	SHA-224, SHA-512/224, SHA3-224	SHA-1
128	SHA-256, SHA-512/256, SHA3-256	
192	SHA-384, SHA3-384	SHA-224, SHA-512/224
≥ 256	SHA-512, SHA3-512	SHA-256, SHA-512/256, SHA-384, SHA-512, SHA3-512

Security strength time frame



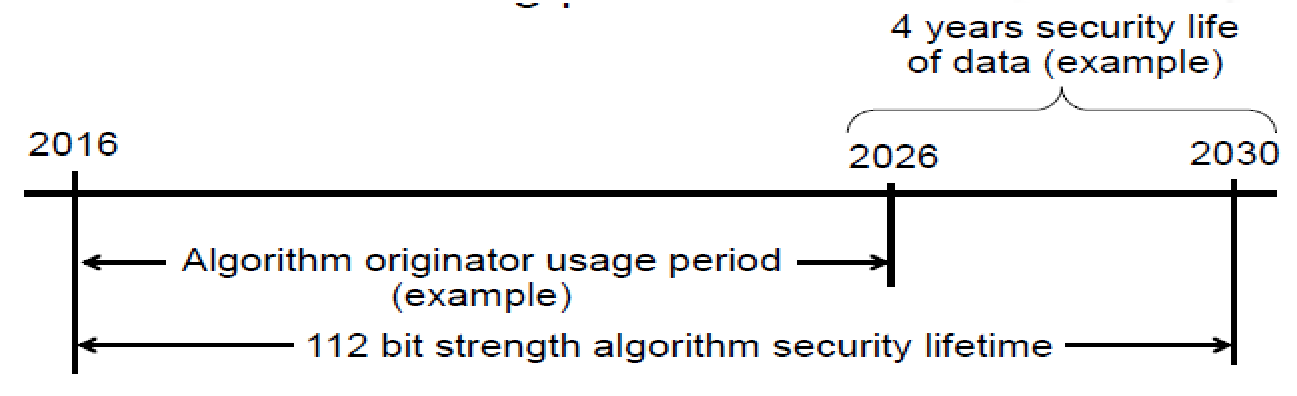
Security Strength		Through 2030	2031 and Beyond
< 112	Applying	Disallowed	
	Processing	Legacy-use	
112	Applying	Acceptable	Disallowed
	Processing		Legacy use
128	Applying/Processing	Acceptable	Acceptable
192		Acceptable	Acceptable
256		Acceptable	Acceptable

Security lifetime

- The estimated time period during which data protected by a specific cryptographic algorithm (and key size) remains secure is called the *algorithm security lifetime.*

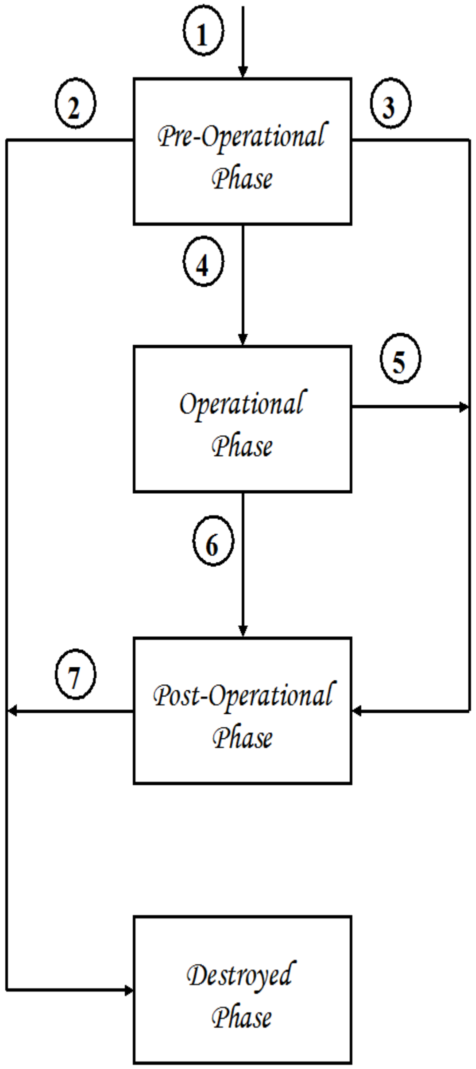
- Typically, an organization selects the cryptographic services that are needed for a particular application. Then, based on the algorithm security lifetime and the security life of the data to be protected, an algorithm and key-size suite is selected that is sufficient to meet the requirements.

- The algorithm security life = (the algorithm originator-usage period) + (the security life of the data beyond the algorithm originator-usage period)


4 years security life of data (example)
2016
2026
2030
Algorithm originator usage period (example)
112 bit strength algorithm security lifetime

key-management lifecycle

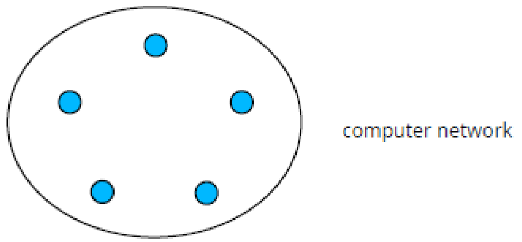
- The cryptographic key-management lifecycle can be divided into four phases.:
  - Pre-operational phase: User registration, Activation date setup, Definition and configuration of pre-shared secrets
  - Operational phase
  - Post-operational phase
  - Destroyed phase



Cryptographic issues

- ❑ Cryptography resolves security problems in public networks.
- ❑ BUT the remaining challenge is key distribution.

Key distribution challenge



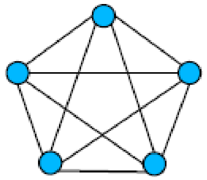
- Assume a computer network (eg the Internet) with *n* nodes
- Each pair of nodes needs a separate key to communicate securely with cryptographic protection
- How many secure key distributions are required?
  - Symmetric secret keys, require **confidentiality**.

    $n(n\text{-}1)/2$ distributions, grow quadratically.

    impractical in open networks.
  - Asymmetric public keys, require **authenticity.**

    $n(n\text{-}1)$ distributions of public keys, grow quadratically

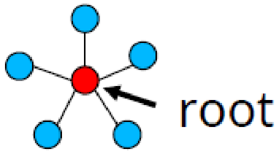
    impractical in open networks



n nodes

$n(n\text{-}1)/2$ edges

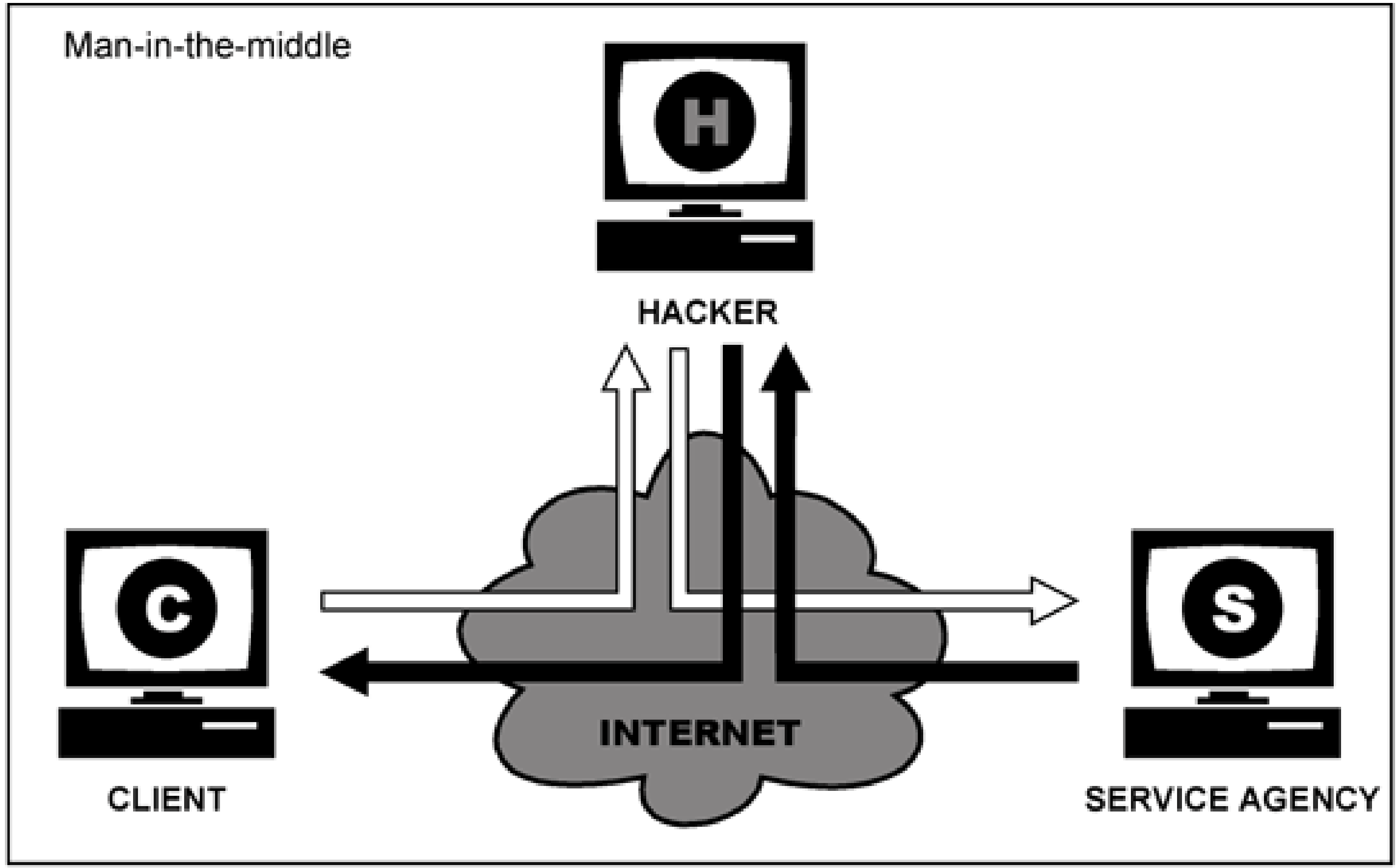
  - Asymmetric public keys with PKI, require **authenticity**.
    - 1 root key is distributed to everyone *n* nodes
    - grows linearly
    - ... much easier, but still relatively challenging



n nodes

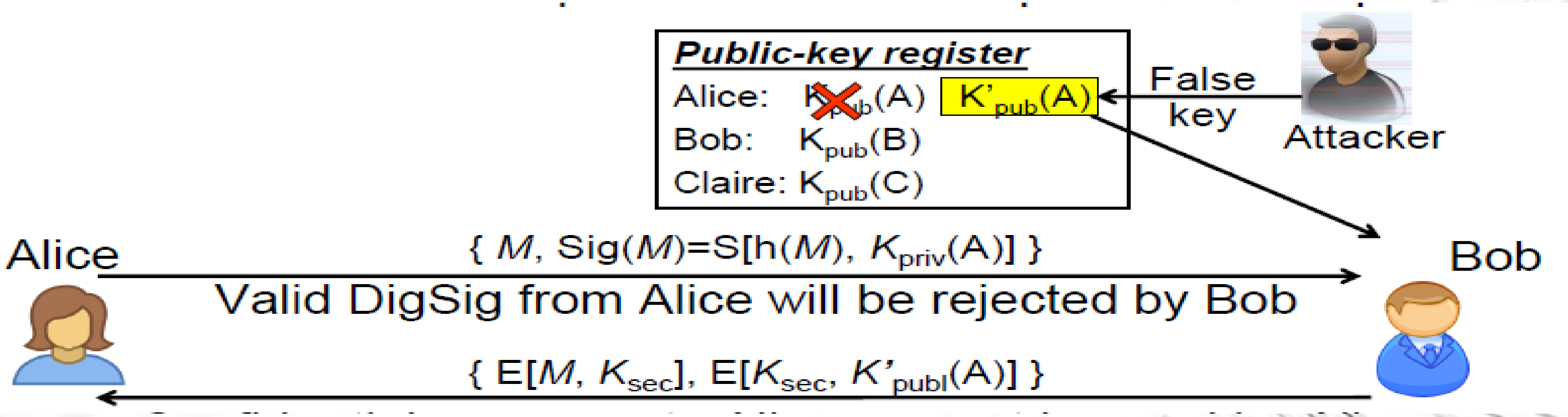
n edges

The problem of key distribution: Man in the Middle attack



The problem of non-authenticated public key

- ❑Pblic Keys are saved on a public repository.
- ❑What would be the consequences if the Attacker repalces the public key of Alice by another key.



Public-key register

Alice: ~~K_{pub}~~(A) K'_{pub}(A)

Bob: K_{pub}(B)

Claire: K_{pub}(C)

{ M, Sig(M)=S[h(M), K_{priv}(A)] }

Valid DigSig from Alice will be rejected by Bob

{ E[M, K_{sec}], E[K_{sec}, K'_{publ}(A)] }

Requirements for Trust

- **Technical solutions** based on cryptography, digital certificates and digital signature.
- **Legal framework**
- **Trusted Third Parties** (certification authorities): security policy and procedures, standards, CP and CPS,…

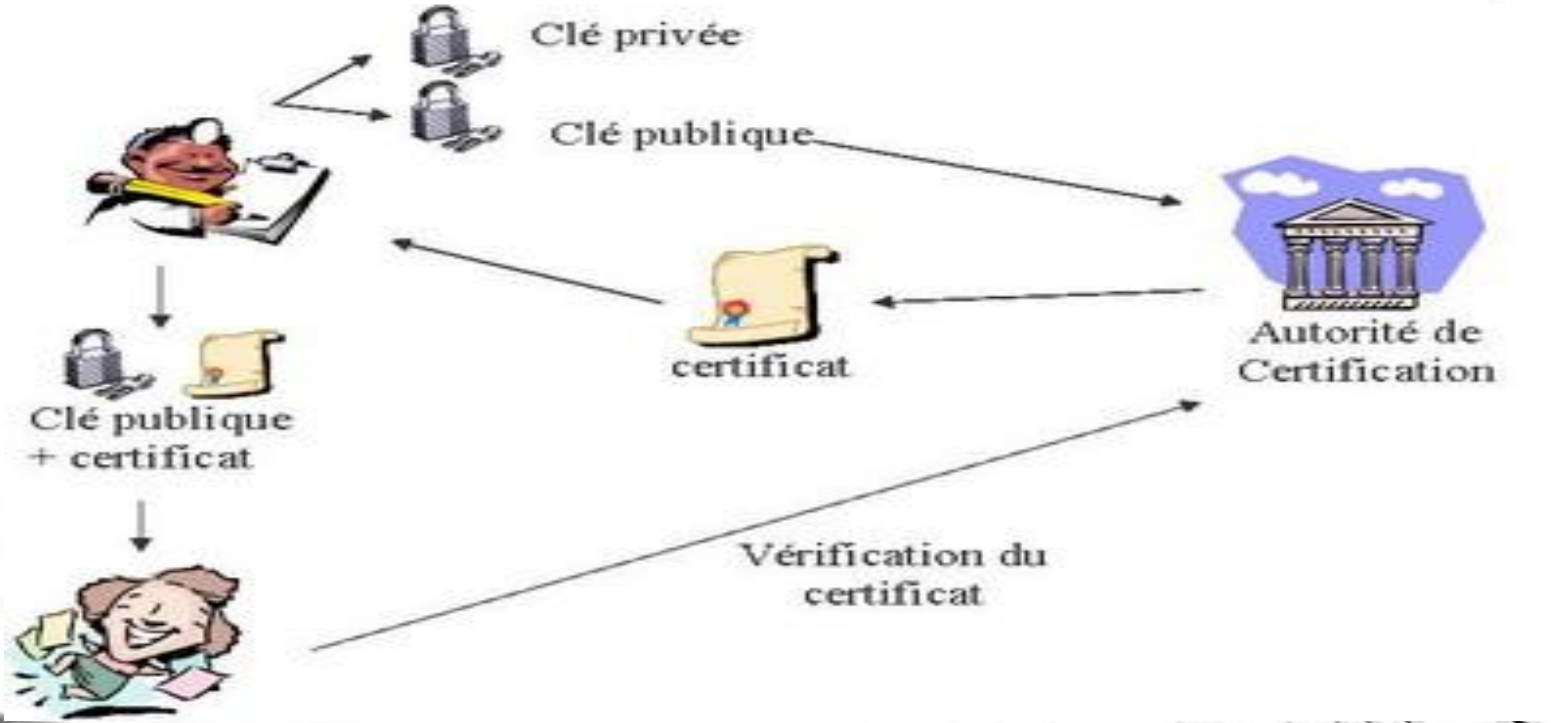
Legal Framework

- Definition of electronic documents, digital certificates and digital signature
- Liability of electronic document digitally signed
- Role and duties of Certification Service Providers (CSP)
- About E-commerce Transactions
- Privacy protection
- Technical data relating to digital certificates and their liability
- Technical specifications for digital signature creation devices

Public Key Infrastructure

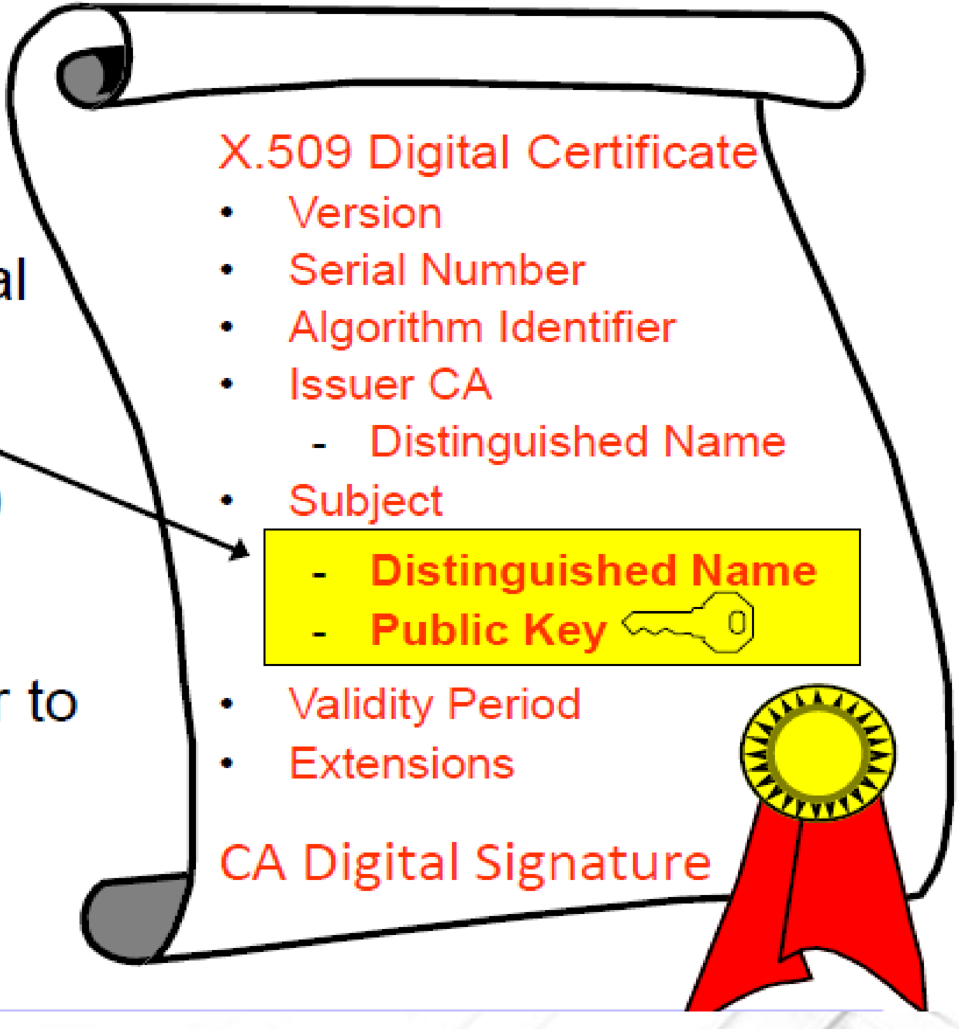
A structure of hardware, software, people, processes and policies that employs digital signature technology to facilitate a verifiable association between the public component of an asymmetric public-key with a specific subscriber that possesses the corresponding private key.

How it is processed?


Clé privée
Clé publique
Autorité de Certification
certificat
Clé publique + certificat
Vérification du certificat

Public Key Certificate

- A public-key certificate is a data record containing a subject distinguished name and a public key with a digital signature by the CA
- Binds name to public key
- Certification Authorities (CA) sign public keys.
- An authentic copy of CA's public key is needed in order to validate certificate
- **Relying party** validates the certificate (i.e. verifies that user public key is authentic)

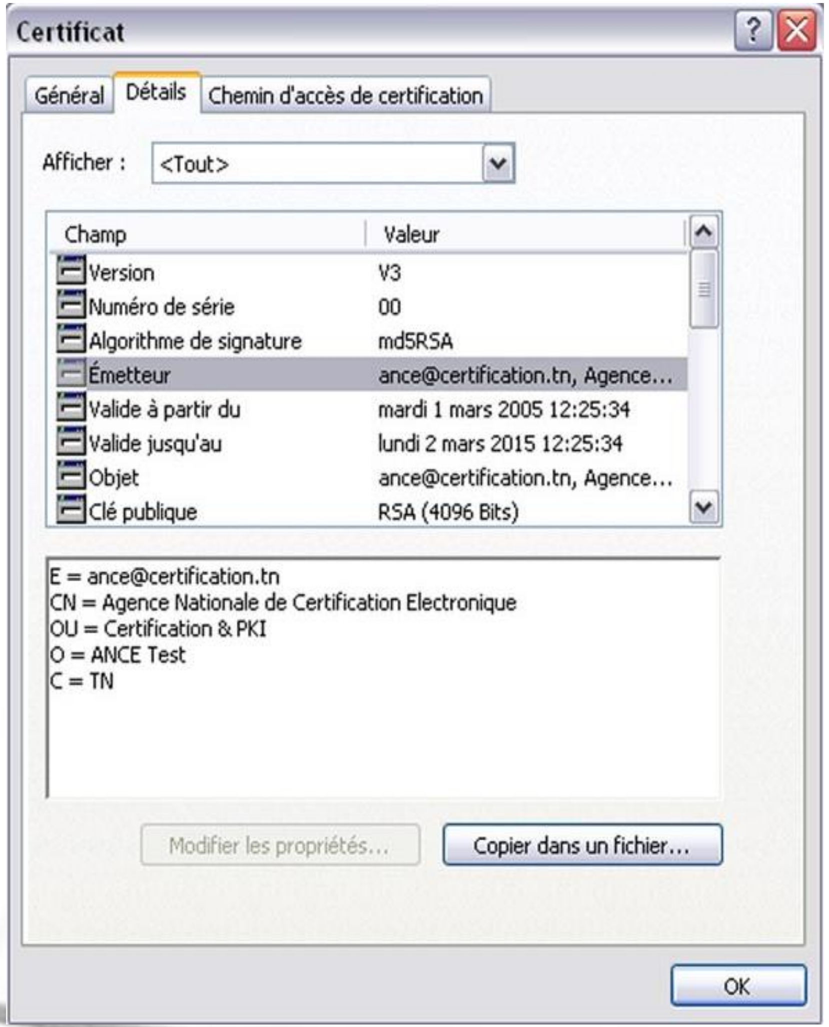


X.509 Digital Certificate

- Version
- Serial Number
- Algorithm Identifier
- Issuer CA
  - Distinguished Name
- Subject
  - **Distinguished Name**
  - **Public Key**
- Validity Period
- Extensions

CA Digital Signature

Digital certificate X.509

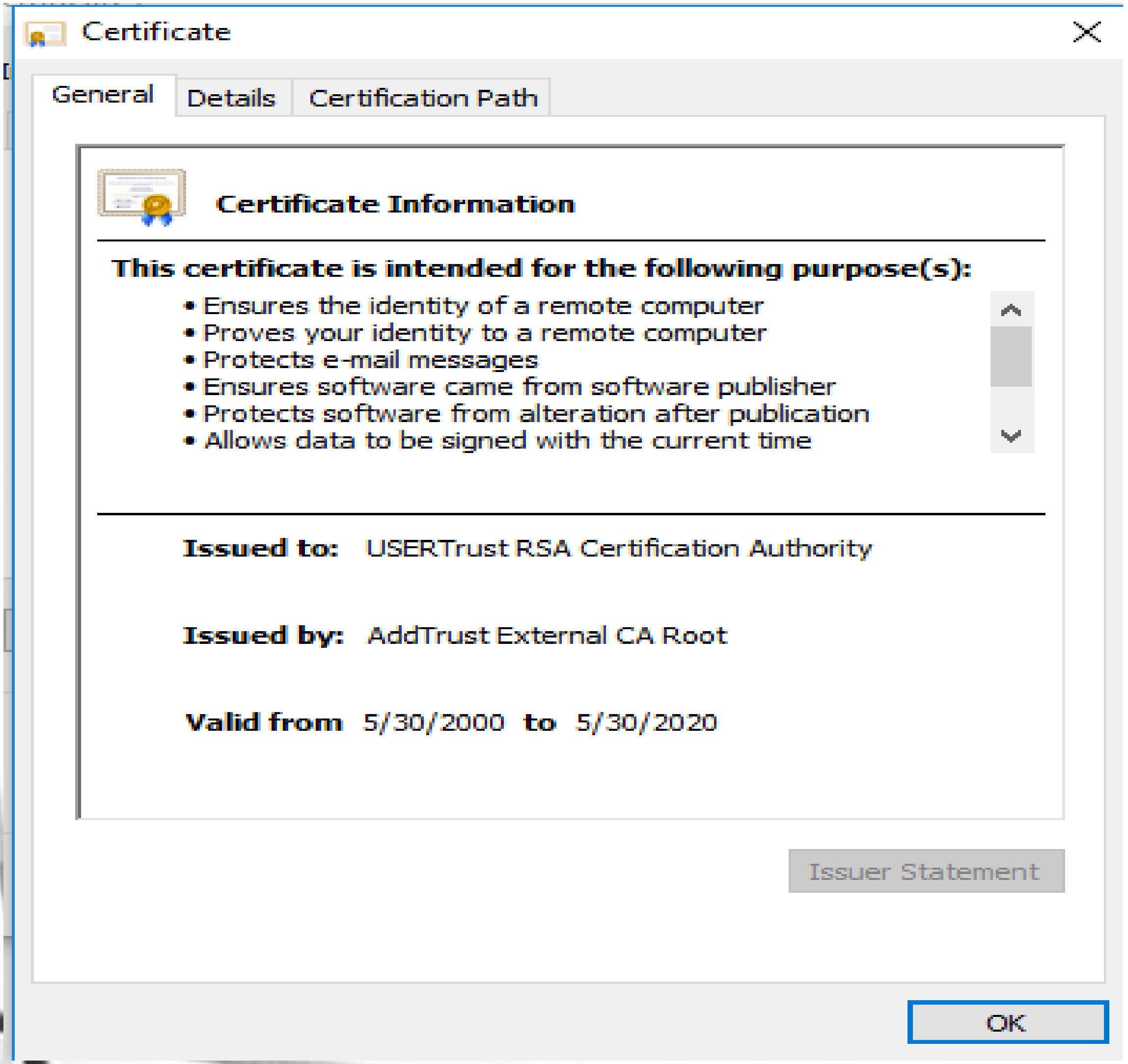

Certificat
Général
Détails
Chemin d'accès de certification
Afficher : <Tout>
Modifier les propriétés...
Copier dans un fichier...
OK

Champ	Valeur
Version	V3
Numéro de série	00
Algorithme de signature	md5RSA
Émetteur	ance@certification.tn, Agence...
Valide à partir du	mardi 1 mars 2005 12:25:34
Valide jusqu'au	lundi 2 mars 2015 12:25:34
Objet	ance@certification.tn, Agence...
Clé publique	RSA (4096 Bits)

E = ance@certification.tn
CN = Agence Nationale de Certification Electronique
OU = Certification & PKI
O = ANCE Test
C = TN

- VERSION
- SERIAL NUMBER (unique)
- SIGNATURE ALGORITHM
- ISSUER: Certification authority
- VALIDITY
- SUBJECT : user name
- SUBJECT PUBLIC KEY INFO
- SIGNATURE

Exemple of an X.509 Certificat



Certificate Information

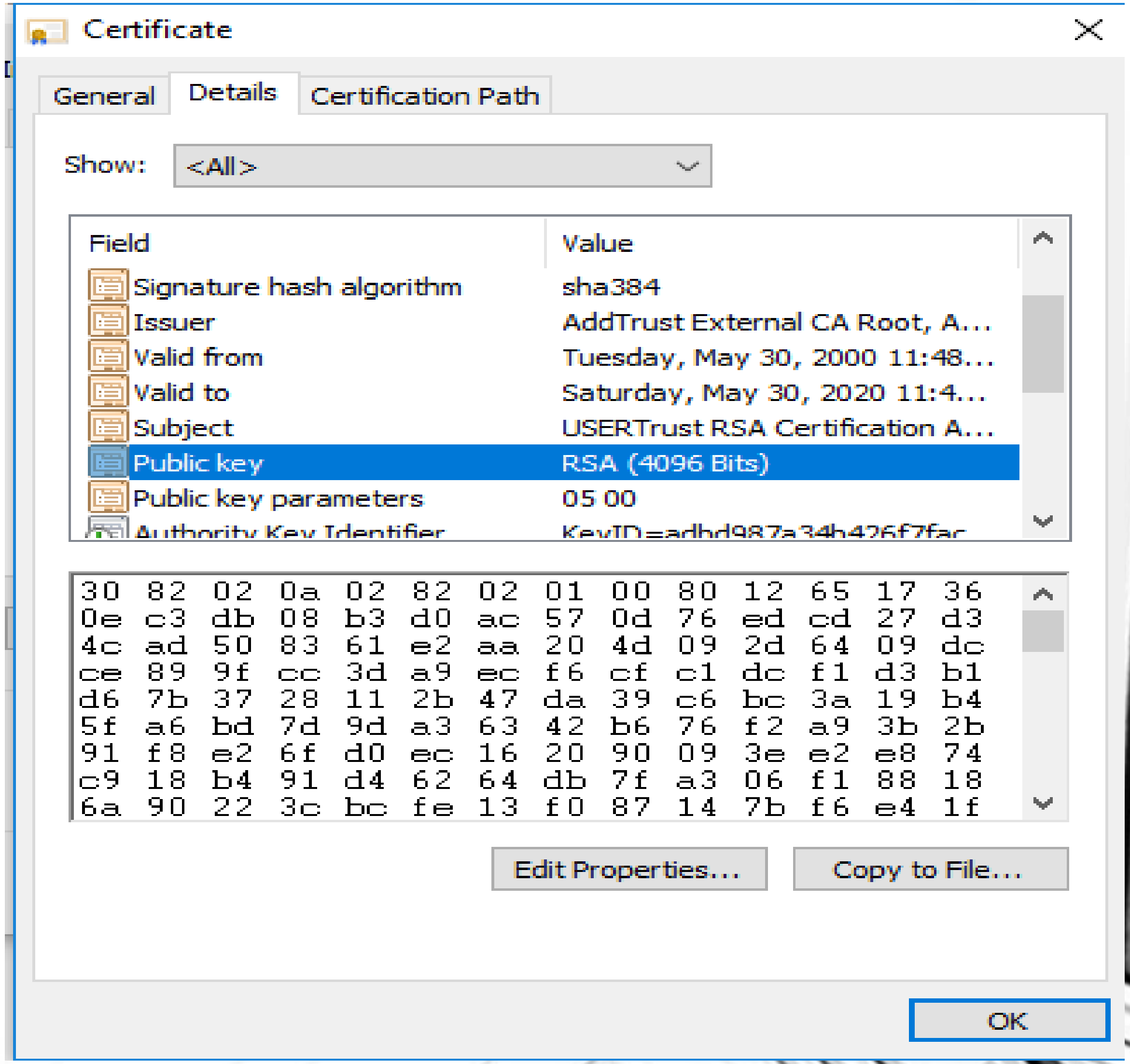
This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- Protects e-mail messages
- Ensures software came from software publisher
- Protects software from alteration after publication
- Allows data to be signed with the current time

Issued to: USERTrust RSA Certification Authority

Issued by: AddTrust External CA Root

Valid from 5/30/2000 **to** 5/30/2020

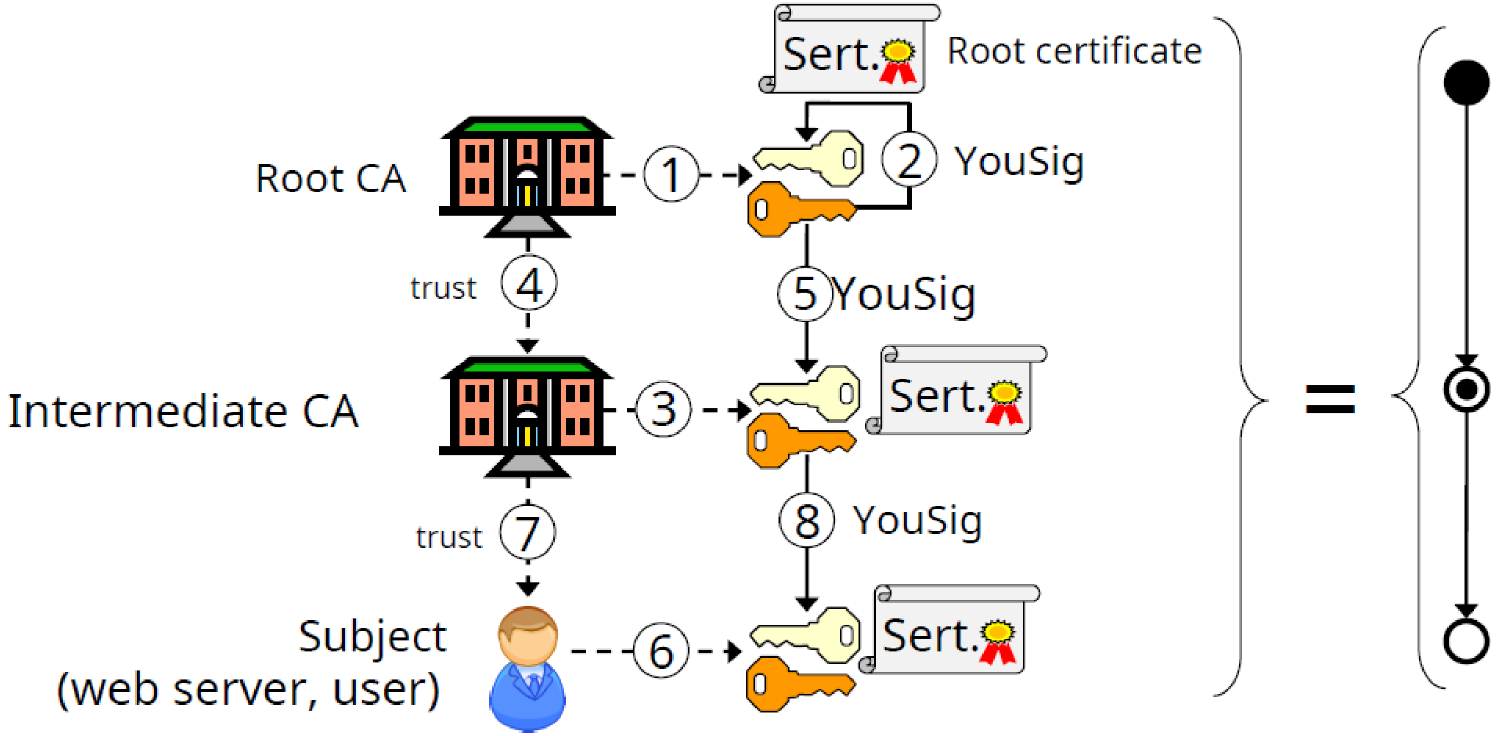


Show: <All>

Field	Value
Signature hash algorithm	sha384
Issuer	AddTrust External CA Root, A...
Valid from	Tuesday, May 30, 2000 11:48...
Valid to	Saturday, May 30, 2020 11:4...
Subject	USERTrust RSA Certification A...
Public key	RSA (4096 Bits)
Public key parameters	05 00
Authority Key Identifier	KeyID=adbd987a34b426f7fac

```
30 82 02 0a 02 82 02 01 00 80 12 65 17 36
0e c3 db 08 b3 d0 ac 57 0d 76 ed cd 27 d3
4c ad 50 83 61 e2 aa 20 4d 09 2d 64 09 dc
ce 89 9f cc 3d a9 ec f6 cf c1 dc f1 d3 b1
d6 7b 37 28 11 2b 47 da 39 c6 bc 3a 19 b4
5f a6 bd 7d 9d a3 63 42 b6 76 f2 a9 3b 2b
91 f8 e2 6f d0 ec 16 20 90 09 3e e2 e8 74
c9 18 b4 91 d4 62 64 db 7f a3 06 f1 88 18
6a 90 22 3c bc fe 13 f0 87 14 7b f6 e4 1f
```

Chain of certificates



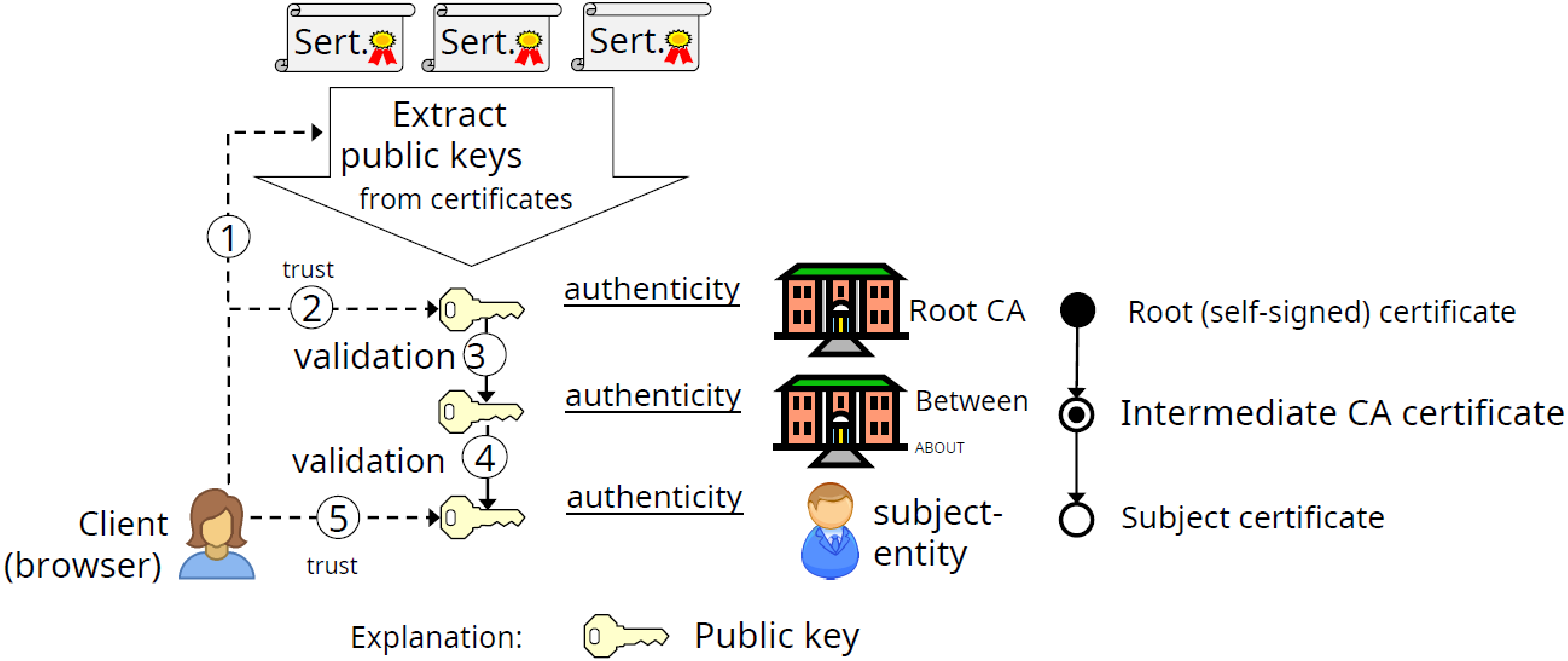
Root certificate issued by root CA. Is normally self-signed and requires secure distribution to ensure authenticity.

Intermediate CA certificate issued by the root CA. Can is validated with a root certificate.

Subject certificate issued by intermediate CA. Can validated with interlying CA certificate.

Explanation: Public key Private key

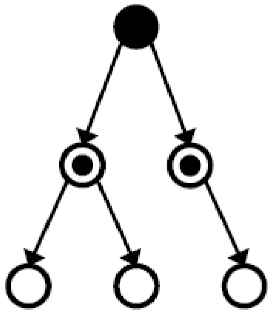
Validation of certificates



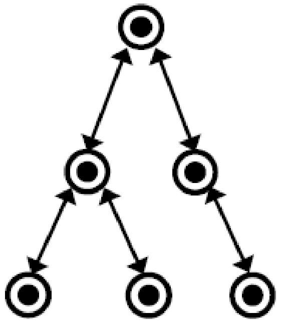
trust-models

Explanation: ● Self-signed root CA certificate CA-
◉ signed intermediate CA cert

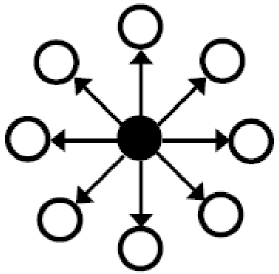
○ CA-signed subject certificate
⦿ Self-signed subject certificate



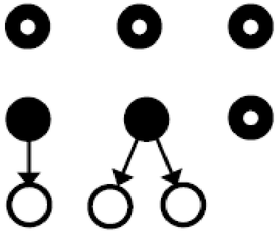
Strict hierarchy



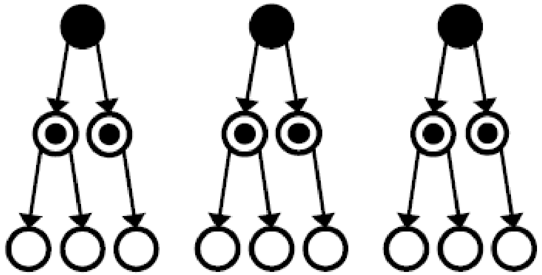
Bi-directional
hierarchy



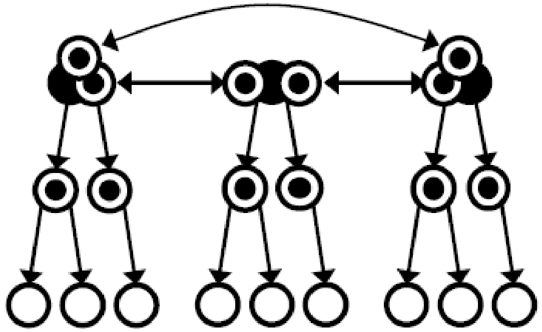
User-centric
(user is his own CA)



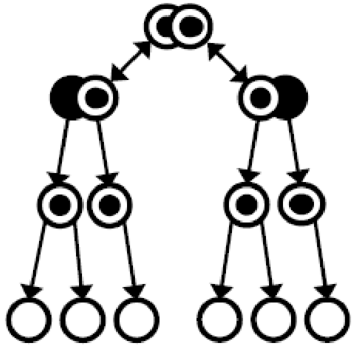
Unstructured PKI



Many isolated strict hierarchies
(Internet PKI)

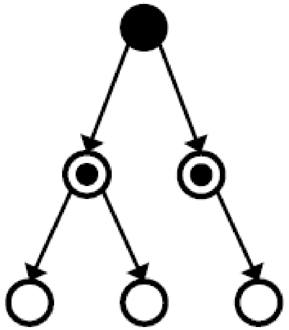


Cross-certified strict hierarchies
(Mesh-PKI)



PKIs with bridge CAs

Strict hierarchy



Explanation:

● Self-signed root CA certificate CA-signed

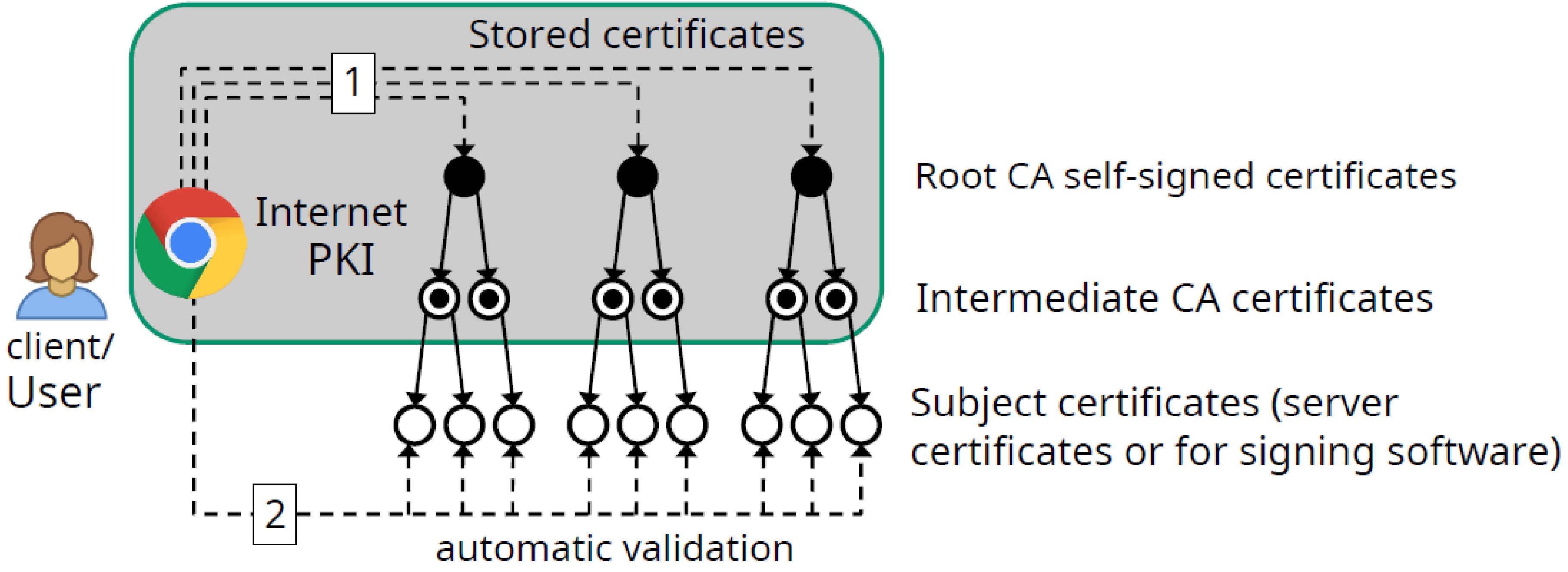
◉ intermediate CA certificate CA-signed

○ subject certificate

- Benefits:
  - - works well in highly structured organizations such as military networks
  - – Simple trust structure
  - - Works well in closed/isolated data networks
- Cons:
  - – All subject entities must trust the same root CA
  - – Compromise of the root CA leads to a total breakdown of security
  - - Does not scale to open data networks

Internet PKI

(PKI used by browsers for https)



Stored certificates
1
Internet
PKI
client/
User
Root CA self-signed certificates
Intermediate CA certificates
Subject certificates (server certificates or for signing software)
2
automatic validation

Internet PKI consists of many isolated strict hierarchies where the (root) CA certificates are installed as part of the browser. New root certificates can be imported by the user or through a software update

Internet PKI and fake certificates

- Certificates are automatically validated by the browser checking the digital signature, and there is a match between the certificate's domain name and the website's domain name
- Criminals can purchase legitimate certificates that are automatically validated by browsers
- Legitimate certificates can be used in conjunction with phishing attacks, e.g. to create a fake website for a bank
- Fake websites may have legitimate certificates !!!
- Server certificate validation is only syntactic authentication, not semantic verification of the website's authenticity
- Users who do not know the server's domain name cannot know in advance whether it is fake or not

Summary

- ❑Public key encryption needs a PKI to be practical
- ❑ PKIs are complex and expensive to operate
- ❑ Internet PKI is the most widely used PKI thanks to the distribution of root certificates with browsers
- ❑ The security of the PKI depends on the integrity of the CAs
- ❑ PKI services are called "Trust Services" in the EU's digital agenda
- ❑ PKI and trust services form the basis for e-ID and e-administration.