

Key Establishment

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
Email: gianluca.dini@unipi.it
Version: 2021-04-28

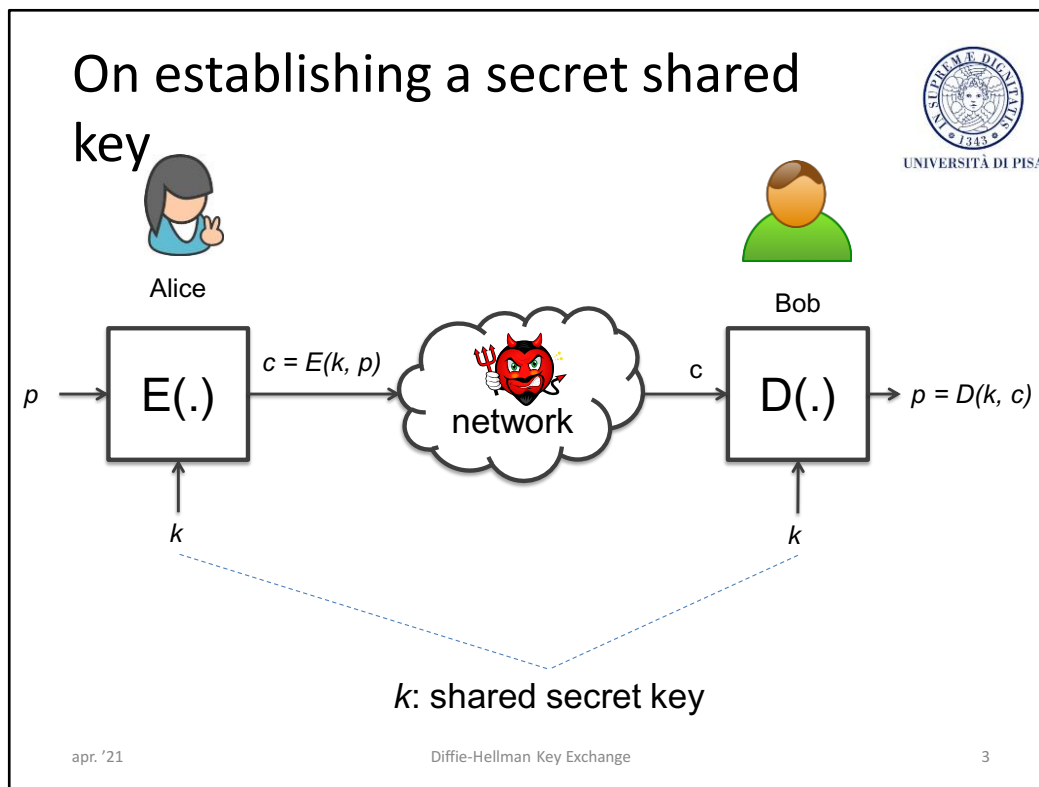
Key Establishment

INTRODUCTION

apr. '21

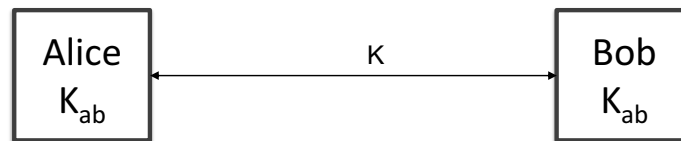
Key establishment

2



Symmetric ciphers assume that Alice and Bob have the same key. Alice and Bob have to agree on the same secret key. There must be a way for Alice and Bob to communicate the key to each other without exposing it. They can't send it over the insecure channel. In the old days, keys were distributed physically (off-line distribution). On the Internet this way of distributing keys is not practical. So, we would like to study the problem of Alice and Bob that start communicating without having a shared key. Key establishment refers to cryptographic protocol that makes it possible to establish a secure shared secret between two or more parties.

Session key



- K_{ab} is a long-term secret shared key
- K is temporary session (ephemeral) key

apr. '21

Key establishment

4

Assume that Alice and Bob share a *long-term secret* key K_{AB} . Typically they do not use it to communicate. They use the key to establish a temporary *session*, or *ephemeral*, key. The approach consists in using a key for a limited amount of time and then updating it. This approach brings about several advantages: 1) Less damage if a key is exposed; 2) Less ciphertext available for analytical attacks; 3) An adversary has to recover several keys if (s)he is interested in decrypting larger parts of plaintext.

Exchanging a session key may be done in several ways.

Session key



UNIVERSITÀ DI PISA

- Key freshness
 - Use a key for a limited amount of time and then update it
 - Session key or ephemeral key
- Advantages
 - Less damage if a key is exposed
 - Less cyphertext available for analytical attacks
 - An adversary has to recover several keys if (s)he is interested in decrypting larger parts of plaintext

apr. '21

Key establishment

5

Session transport and agreement



UNIVERSITÀ DI PISA

- One-pass Key transport
 - M1 $A \rightarrow B: E(K_{ab}, K || t_a)$
 - Where t_a is a timestamp and equires clock synchronization
- Key transport with challenge-response
 - M1 $B \rightarrow A: n_b$
 - M2 $A \rightarrow B: E(K_{ab}, K || n_b)$
 - where n_b is a nonce, i.e., a fresh quantity never used before

apr. '21

Key establishment

6

Assume that Alice and Bob share a *long-term secret* key K_{AB} . Typically they do not use it to communicate. They use the key to establish a temporary *session*, or *ephemeral*, key. The approach consists in using a key for a limited amount of time and then updating it. This approach brings about several advantages: 1) Less damage if a key is exposed; 2) Less cyphertext available for analytical attacks; 3) An adversary has to recover several keys if (s)he is interested in decrypting larger parts of plaintext.

Exchanging a session key may be done in several ways.

Session key



- Key agreement
 - M1 $B \rightarrow A: n_b$
 - M2 $A \rightarrow B: E(K_{ab}, K' || n_a || n_b)$
 - M3 $B \rightarrow A: E(K_{ab}, K'' || n_a)$
 - Where n_a and n_b are nonces and $K = f(K', K'')$
 - Examples of $f()$:
 - 1) $K = K' \oplus K''$
 - 2) $K = H(K' || K'')$, with $H(\cdot)$ secure hash function

apr. '21

Key establishment

7

Assume that Alice and Bob share a *long-term secret* key K_{AB} . Typically they do not use it to communicate. They use the key to establish a temporary *session*, or *ephemeral*, key. The approach consists in using a key for a limited amount of time and then updating it. This approach brings about several advantages: 1) Less damage if a key is exposed; 2) Less cyphertext available for analytical attacks; 3) An adversary has to recover several keys if (s)he is interested in decrypting larger parts of plaintext.

Exchanging a session key may be done in several ways.

Terminology



- Key Establishment
 - Key Transport
 - One party generates and distributes secret key
 - Key Agreement
 - Parties jointly create a secret key
- Key Establishment is strongly related to identification

Key update



- Run the key establishment protocol
 - Performance impact
- Key derivation function
 - Nonce: random number, counter
 - Cipher or MAC

Key update – random number



UNIVERSITÀ DI PISA

Alice (K_{AB})Bob (K_{AB}) $\text{rnd} = \text{RNG}()$

< ----- rnd -----

derive key

derive key

 $K_{\text{ses}} = E_{K_{AB}}(\text{rnd})$ $K_{\text{ses}} = E_{K_{AB}}(\text{rnd})$

- HMAC can be used instead of E
- A counter saves the message but requires clock synchronization $K_{\text{ses}} = E_{K_{AB}}(\text{cnt}++)$

apr. '21

Key establishment

10

The n^2 Key Distribution Problem



UNIVERSITÀ DI PISA

- n users where each party securely communicates with everyone
- Each pair of users shares a long-term secret pairwise key
 - Key pre-distribution
 - Out-of-band transmission

apr. '21

Key establishment

11

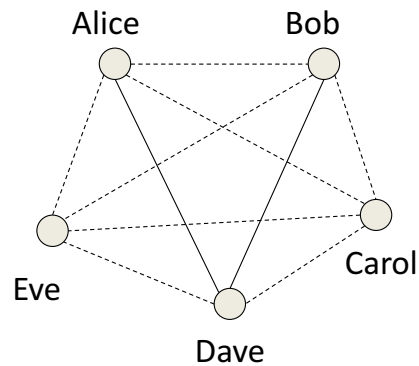
In a pairwise key management scheme, every pair of users share a pairwise key. It follows that Every user has to store $n - 1$ keys. The total number of keys is: $(n-1)+(n-2)+\dots+1 = n(n-1)/2$ that is in the order to n^2 .

The n^2 Key Distribution Problem



UNIVERSITÀ DI PISA

- Every user stores $(n - 1)$ keys
- There are $\binom{n}{2} = \frac{n \cdot (n-1)}{2}$ symmetric key pairs in the system which is in the order of n^2 .



apr. '21

Key establishment

12

In a pairwise key management scheme, every pair of users share a pairwise key. It follows that Every user has to store $n - 1$ keys;
The total number of keys is: $(n-1) + (n-2) + \dots + 1 = n(n-1)/2$ that is in the order to n^2 .

The n^2 Key Distribution Problem



- Pros: Security
 - If a subject is compromised only its communications are compromised; communications between two other subjects are not compromised
 - We cannot do any better!
- Cons: Poor scalability
 - The number of keys is quadratic in the number of subjects
 - A new member's joining/leaving affect all current members

apr. '21

Key establishment

13

The pairwise scheme has pros and cons. In terms of security, if a subject is compromised only its communications are compromised. The remaining users can keep communicating. However, the scheme scales poorly.

The n^2 Key Distribution Problem



- Pre-distribution does not work for large dynamic networks
- Pre-distribution works for small networks where the number of users does not change frequently
 - E.g., branches of a company

apr. '21

Key establishment

14

The pairwise scheme has pros and cons. In terms of security, if a subject is compromised only its communications are compromised. The remaining users can keep communicating. However, the scheme scales poorly.

Key Establishment

KEY ESTABLISHMENT USING SYMMETRIC-KEY TECHNIQUES

apr. '21

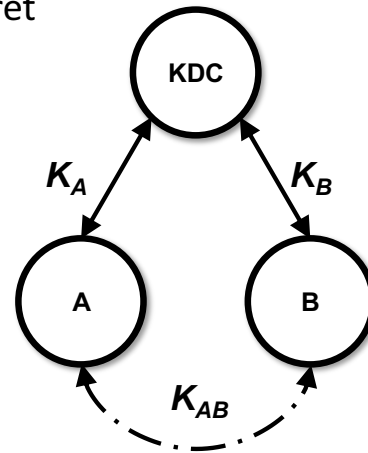
Key establishment

15

Key Distribution Center



- Each user shares a long-term secret key with KDC
 - Key Encryption Key (KEK)
- Each KEK constitutes a secure channel
- KEKs are pre-distributed



apr. '21

Key establishment

16

In a TTP-based scheme, TTP is a *trusted third-party* that

1. Maintain a database $\langle U, K_U \rangle$
2. Guarantee integrity and secrecy of the database
3. Correctly play *the key management protocol*

In addition, Each user shares a *long-term, a priori* key with TTP. The overall number of long-term keys is $O(n)$. TTP allows a pair of users to establish a session key (*key management protocol*).

When a new member joins (or leaves), we have to update only the TTP (add or delete the member's key: just one key). If a member is compromised, then only its communications are compromised. However, if the TTP is broken the whole system is broken.

Performance and security issues



UNIVERSITÀ DI PISA

- Performance: better scalability than pairwise scheme
 - The overall number of KEKs is n
 - Each user stores 1 KEK
 - Each member's joining/leaving only 1 KEK has to be established/removed
- Security
 - If a user is compromised, its communications are compromised
 - If KDC is compromised, all communications are compromised

apr. '21

Key establishment

17

Key Distribution Center



- KDC is a single point of failure
 - Performance
 - KDC must be available
 - KDC must be efficient
 - Security
 - KDC knows all the keys
 - KDC can read all msg between Alice and Bob
 - KDC can impersonate any party
 - KDC must a trusted third party

apr. '21

Key establishment

18

Basic KE using KDC (1/2)



UNIVERSITÀ DI PISA

Alice	KDC	Bob
KEK: K_A	KEK: K_A, K_B	KEK: K_B
-- REQ(A, B) -- >		
	$K_{AB} \leftarrow \text{RNG}()$	
	$Y_A = E_{K_A}(K_{AB})$	
	$Y_B = E_{K_B}(K_{AB})$	
< ---- Y_A ---- ---- Y_B ---- >		
$k_{AB} = D_{K_A}(Y_A)$		$K_{AB} = D_{K_B}(Y_B)$
< ---- $E_{K_{AB}}(\text{session})$ --- >		

apr. '21

Key establishment

19

The Key Encryption Keys (KEKs) are long-term keys used to encrypt other keys or secrets.

Basic KE using KDC (2/2)



UNIVERSITÀ DI PISA

Alice	KDC	Bob
KEK: K_A	KEK: K_A, K_B	KEK: K_B
-- REQ(A, B) -- >		
	$K_{AB} \leftarrow \text{RNG}()$	
	$y_A = E_{K_A}(K_{AB})$	
	$y_B = E_{K_B}(K_{AB})$	
	< ----- y_A, y_B ---->	
$k_{AB} = D_{K_A}(y_A)$		
----- y_B ----- >		
		$K_{AB} = D_{K_B}(y_B)$
	< ----- $E_{K_{AB}}(\text{session})$ ----- >	

apr. '21

Key establishment

20

AI alternative communication scheme. From the security standpoint, the tow protocols are equivalent because it is the KDC that communicates witj Alice and Bob by means of y_A and y_B . This protocol is simplified and suffers from two problems: replay attack and key confirmation attack.

Security issues



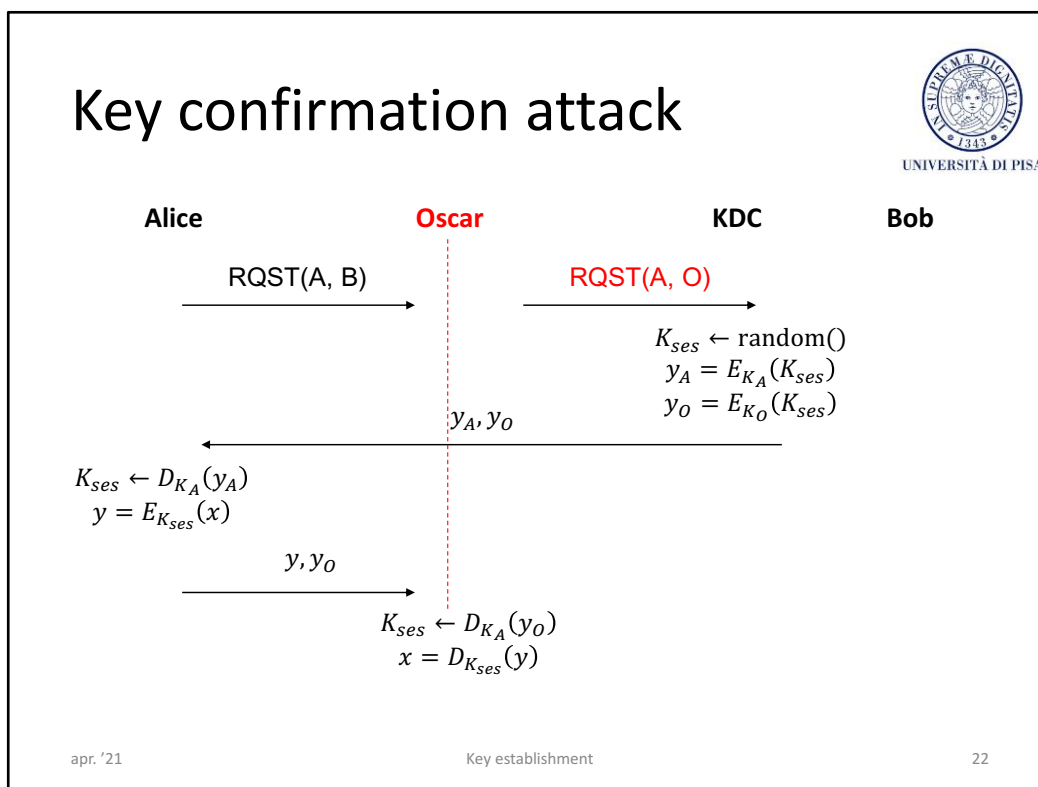
UNIVERSITÀ DI PISA

- **Replay Attack**
 - The adversary records the key establishment protocol
 - The adversary replays y_A and/or y_B
 - The adversary make users to use an old session key
 - An old session can be replied (the session has to be recorded)
 - A compromised session key can be reused
- **Key Confirmation attack**
 - MIM attack performed by a legitimate but malicious user

apr. '21

Key establishment

21



Assume that Oscar (the adversary) is able to control Alice's communications.

FNO A Qui
SU UZUat 11

Key establishment techniques

USING ASYMMETRIC TECHNIQUES

apr. '21

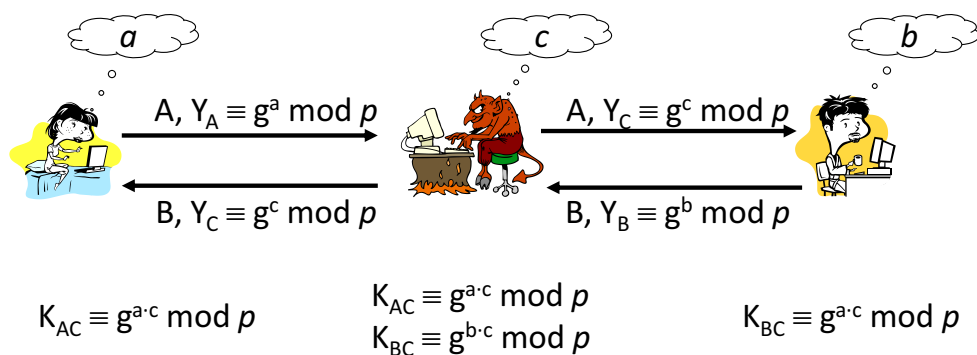
Key establishment

23

Man-in-the-middle Attack



UNIVERSITÀ DI PISA



apr. '21

Key establishment

24

We know that plain DHKE protocol suffers from the man-in-the-middle attack.

Certificate



- Certificate

- Data structure that cryptographically links the identifier of a subject to the subject public key (and other stuff):

$$\text{Cert}_A = A, \text{pubK}_A, L_A, S_{CA}(A || \text{pubK}_A || L_A)$$

- A: identifier; pubK_A : public key; L_A : validity interval; || concatenation operator
- Certification Authority (CA) is a **TTP** that attests the authenticity of a public key
- CA's signature **indissolubly links** identifier and public key (and other parameters)

apr. '21

Key establishment

25

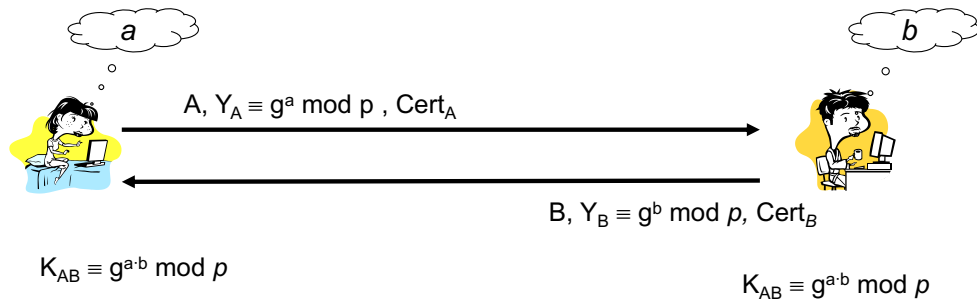
A certificate is a sort of digital identity card

We played a trick: the problems with users' pubKs have been moved onto CA's pubK. So, where are CA's pubK published? In browsers. Is this secure? No! A virus can change the CA's certificates DB

Man-in-the-middle Attack



UNIVERSITÀ DI PISA



apr. '21

Key establishment

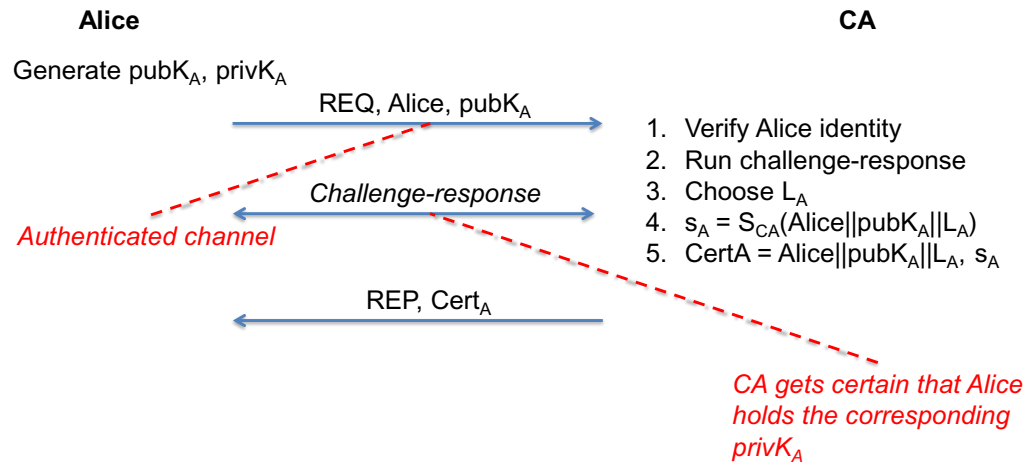
26

Certificate generation



UNIVERSITÀ DI PISA

User-provided Keys



apr. '21

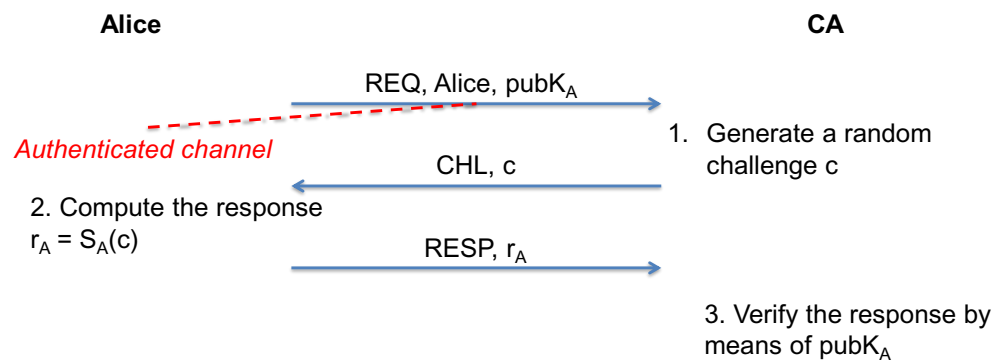
Key establishment

27

Challenge-response protocol



User-provided Keys

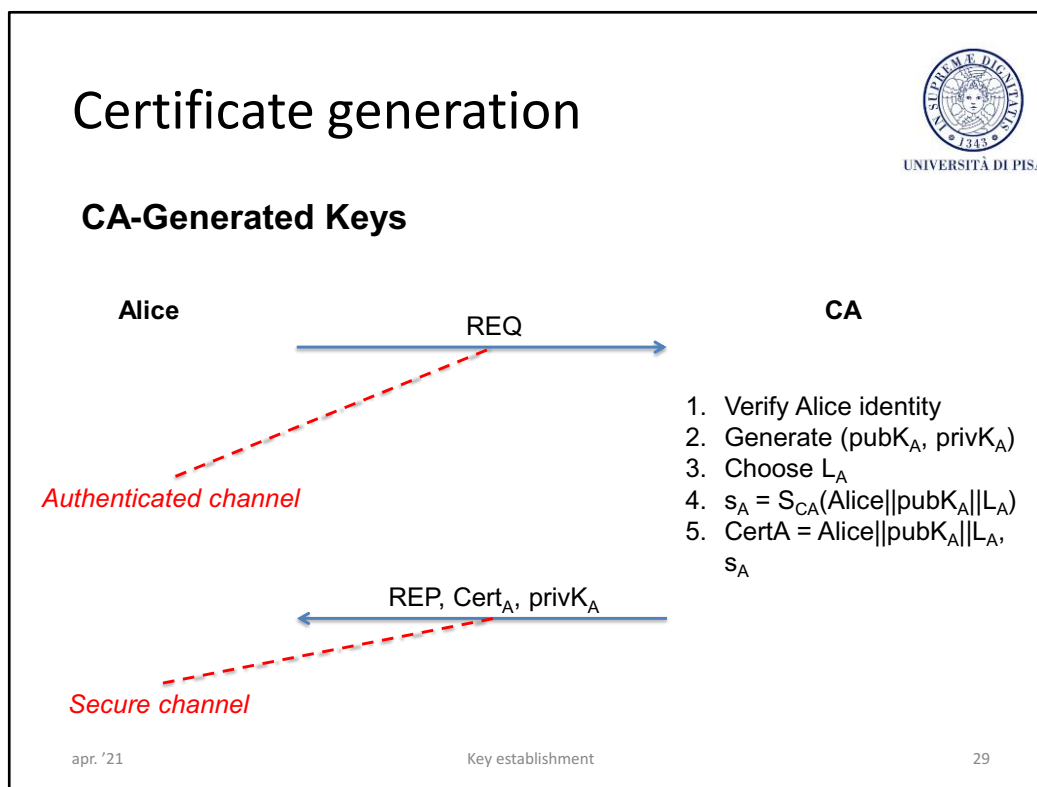


- CRP can be implemented also with a cipher

apr. '21

Key establishment

28



Notice that the privK must not be available to the CA generator.

On key generation at CA-side



UNIVERSITÀ DI PISA

Fatal crypto flaw in some
government-certified smartcards
makes forgery a snap
(www.arstechnica.com)



- Fatal flaw in the hw RNG
- Smartcards passed two international certifications
- Research paper at AsiaCrypt 2013

apr. '21

Key establishment

30

The team of scientists uncovered what their paper called a "fatal flaw" in the hardware random number generator (RNG)

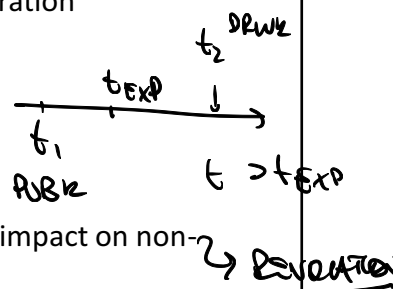
The vulnerable smartcards used in the Taiwanese program passed the FIPS 140-2 Level 2 and the Common Criteria standards. The certifications, managed by the National Institute of Standards and Technology (NIST)

Backup of private key



UNIVERSITÀ DI PISA

- Public key encryption
 - Backup privK or encrypted data may become inaccessible
 - Be able to decrypt even after key lifetime expiration
 - Government backs up of citizen's privK
 - This raises privacy issues
 - Company backup of employee's privK
 - Encrypted data belong to the company
- Digital signature?
 - Delete the key after key expiration or adverse impact on non-repudiation
 - Expensive recovery in large scale apps as you have to redistribute the pubK
 - Threshold crypto (t out of n)



apr. '21

Key establishment

31

Backup of decryption privK. Actually, if you lose privK then you can't decrypt your encrypted data. Government wants the key to eavesdrop communication whereas companies want keys for recovering stored encrypted data. Actually, companies can eavesdrop communication at the end-points.

Backup of a signing privK is a completely different matter. Of course if you backup a signing privK, then you can claim that someone took your backup and signed on your behalf. So backup of a privK has an adverse impact on non-repudiation. For example, German law forbids backup of a signing key. However, if you run a large-scale application, in case you lose the privK, you have to redistribute the corresponding pubK to many (millions) of users/devices. For instance, if you are Microsoft or Adobe, your software embeds the pubK which verifies the integrity and authenticity of a software update. As a further example, if you are a financial application, e.g. Visa, then your pubK is in millions of terminals.

That's why you want different key pairs for encryption and signing.

VALLO PREVENIRE A SINGOLA DOTT OF FAILURE

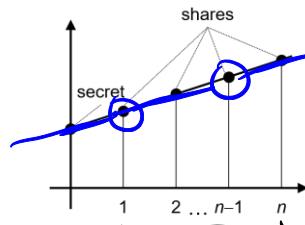
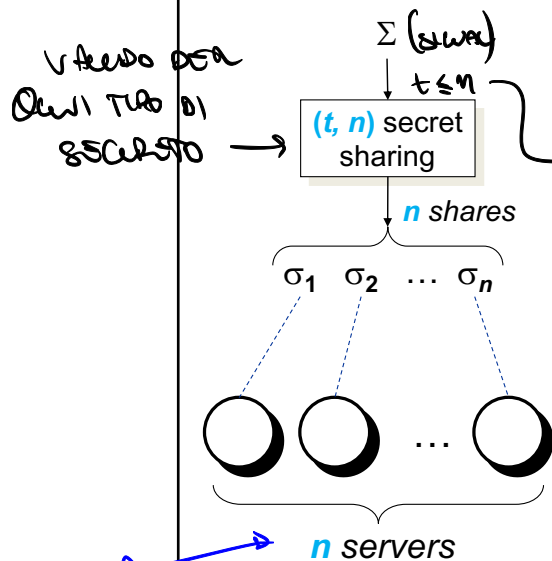
Threshold crypto (intuition)



UNIVERSITÀ DI PISA

SECRET SHARING

- The secret (private key Σ) is split into n shares
- At least t shares are necessary to reconstruct the secret
- The system tolerates the compromise of $t-1$ nodes



$(2, n)$

MESSAGGIO

ME SCONO 2 PER RECONSTRUIRE

Polynomial $(2, n)$ secret sharing

n PARTI = SHARES

32

AUMENTA LA POSSIBILITÀ
CHE UN SERVER VENGA
COMPROMESSO

PER COSTRUIRE LA CUNDA BASTANO
2 SHARES

Really, threshold crypto is more general than this. For example, each server can generate a share of a signature. t -shares of a signature are sufficient to reconstruct the whole signature. Threshold crypto is 90s technology developed by CertCo and sold to Visa. It is an example of good technology that is not used.

CA's obligations



- CA must be reliable
 - CA must verify that the name (Alice) goes along with the key (privKA)
 - CA must verify that owner of (privK, pubK) pair is really entitled to use that name
 - CA establishes rules/policies to verify that a person has rights to the name
 - Identifying a subject is not easy; depends on country

apr. '21

Key establishment

33

CA must be reliable. As a leading example, assume that Mallet wants to acquire a certificate that links its public key to Uncle Scrooge McDuck. If he manages to do so, any digital signature he makes could be ascribed to Uncle Scrooge. As a further example, consider the case that Mallet asks for a certificate identifying him as Vice-President of Art at ACME Corporation.

Verify the identity of a subject. Napoleon needed soldiers. Everyone took a name and Napoleon made a public register. In France, identity card was established in 1940 during the Vichy Republic to support Olocausto. In Italy, identity cards were established during the Fascism period so that it was possible to monitor dissidents (TULPS – Testo Unico delle Leggi di Pubblica Sicurezza, art.4). Changing the name was against law. Changing name is very difficult in every part of Europe except for anglo-saxons. In UK/US changing name is easy, unless you are involved in a trial. During the trial you can't. In UK if you want to open a bank account you are asked the gas/electricity bill to identify you. In UK/US, CA just believe you (they ask you the electricity/gas bill). In US/UK checking whom somebody actually is very hard, harder than in the rest of Europe. As a consequence, given these difficulties in identification, a CA gets liable of almost nothing.

CA's obligations



- CA's certificate must be (immediately) available
 - CA's certificate is released at user registration time
 - CA's certificate is published in newspapers
 - CA's certificate is embedded in a browser installation package (is this secure?)

apr. '21

Key establishment

34

CA's certificate must be available.

Writing the CA list in browsers is not secure. A virus can update this list. Furthermore, the list is difficult to be centralised controlled by a sysadmin because a user can change this list individually.

Trust delegation



UNIVERSITÀ DI PISA

- Certification is based on trust delegation (trust transfer)
 - Bob trusts and delegates CA to verify Alice's identity and attest the authenticity of pubK_A
 - Bob trusts the authenticity of CA's pubK_{CA}
 - then
 - Through a certificate Cert_A signed by CA, Bob acquires trust (believes) in the authenticity of pubK_A

Important to remember



UNIVERSITÀ DI PISA

- A certificate defines an indissoluble link between a subject's identifier and public key
- A certificate does not specify the meaning of that link
- A certificate doesn't specify the possible uses of that key → *non sono legati*
- A certificate doesn't make any statement on the trustworthiness of the subject

apr. '21

Key establishment

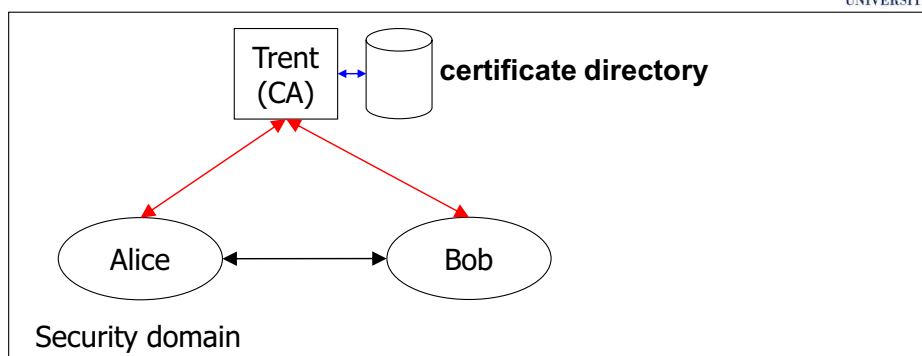
36

non è un problema della CA

Single CA Model



UNIVERSITÀ DI PISA

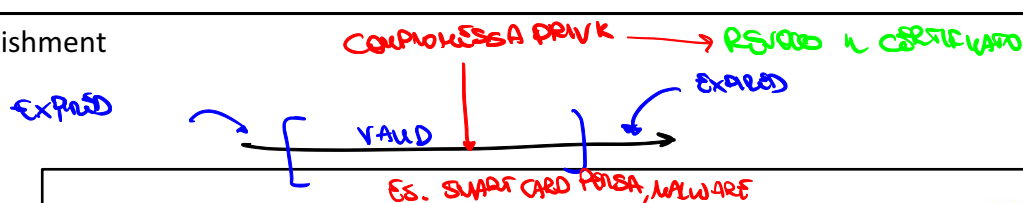


- **Security domain** under control of the CA
- **Certificate directory** is a read-only database that stores certificates

apr. '21

Key establishment

37



Expired & revoked certificates



UNIVERSITÀ DI PISA

- A certificate is **expired** if the validity period is expired
- If the private key gets compromised before expiration, then the certificate must be **revoked**
 - The private key has been revealed
 - The subject has changed role or left the organization
- Certificate revocation must be
 - Correct: revocation can be granted only to authorized parties, i.e., **the owner** or **the issuer**
 - Timely: revocation must be disseminated to all interested parties as soon as possible

28/04/2021

Public Key Infrastructures

38

How to verify a certificate

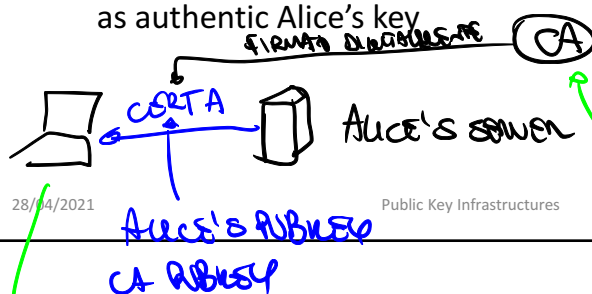


UNIVERSITÀ DI PISA

• Bob's verification of Alice's Cert_A

1. Bob obtains CA's public key eT [once at set-up]
 2. Bob verifies validity of CA's public key [once at set-up]
 3. Bob verifies the digital signature in Cert_A by using pubK_{CA}
 4. Bob verifies that Cert_A is valid
 5. Bob verifies that Cert_A is not revoked
- If all these checks are successful, then Bob accepts pubK_A as authentic Alice's key

SECRET
CERTIFICATE



28/04/2021

Public Key Infrastructures

39

PER VERIFICARE LA
PUBBLIC KEY DI UNA
DIGITALE DELLA CERTIFICATION AUTHORITY

DA FARE SOLO
AL SETUP DEL
SISTEMA -
AD OGNI USUO DEL
CERTIFICATO!

COME VERRE SE UN CERTIFICATO E' STATO REVOCATO O NO?

Revocation options



UNIVERSITÀ DI PISA

- Certificate Revocation List
 - offline (CRL)
- Online Certificate Status Protocol
 - online (OCSP)

CA DOVREBBE AVERE
ENTRAME

apr. '21

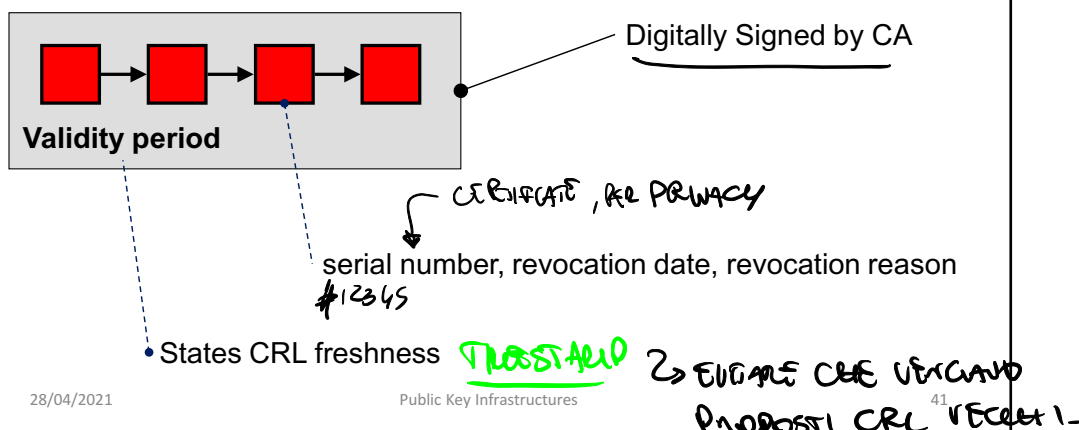
Key establishment

40

CRL



- A CRL is published periodically
- A revoked certificate lies in CRL until expiration
- Δ -CRL for efficiency



A CRL contains the serial numbers of certificates, neither public keys owners nor names, for privacy reasons. A certificate is revoked when an employee is fired. A company does not want to publish the names of the people it fires or those who left organizations. Companies don't like to publish roles and hierarchies of the company. So it is not simple to have a certificate email.

Often CRL are the last component to be realized. Often they are not even realized. Earlier browser didn't even implement CRL access.

A CRL is published periodically (every day, every hour). Remember that in the 90s, people had small bandwidth. They had kbytes. They connected through modems. Furthermore, flat rates didn't exist, you were billed time-wise (per per-second) or traffic-wise (per-byte). Downloading a CRL that is Mbytes was too expensive. They simply didn't do it. So browser didn't check whether a certificate had been revoked. There is a nice story about this.

The Microsoft – Verisign case – Second episode. Upon attempting to access to Verisign CRL, Microsoft realized that certificates released by the latter didn't even have the pointer to CRL, that is the CRL Distribution Point (CRLDP) field was not set. Microsoft implemented the following solution.

Solution #1. Microsoft released an update that

1. Installs a CRL onto the local machine. The CRL lists the two bogus certificates as having been revoked.
2. Adds new functionality via a piece of software called an *installable revocation handler* that causes the system to check the CRL on the machine if the CDP data is missing or invalid.
3. Turns on CRL checking in IE for software publisher's certificates.

Solution #2. If a user doesn't want to install the update, the following solution is also possible

1. **Assumption.** First of all, Microsoft adopts a policy according to which Microsoft certificates are not trusted by default. Trust is given on a per-certificate basis and not on a per-name basis. That is, you can specify that you trust a particular certificate because the name on it is Microsoft, but there isn't any way to say that you want to trust **all** certificates that say Microsoft on them.
2. When a new Microsoft certificate is encountered, a warning dialogue is displayed asking whether to trust content signed by Microsoft, visually inspect the certificate and verify that it isn't one of the bogus ones.
3. You'll always see a warning dialogue anytime a web site or e-mail tries to run a program that's been signed by a certificate that isn't known to the system - even if the certificate says it's owned by Microsoft. Of course, if you say to trust the certificate, you won't see the warning again.

Notes

- Of course if you install the update (solution #1) you will not see the warning dialogue for the bogus certificates.
- A certificate's revocation status takes precedence over its trust status. This means that even if you've previously said you trust one of the certificates, the update would display a message telling that it was untrustworthy if you subsequently encountered something signed using it. However, it would be too late. Actually, if you've agreed to trust either of the two certificates, it's very likely that you allowed a program signed using the certificate to run on your system. It hardly matters that the update would alert you if you tried to subsequently run a program signed using the same certificate, because your system could have been completely compromised by the first program.

OCSP



- Protocol sketch
 - Alice → OCSP: <OCSP RQST, Bob's cert serial nr.>
 - OCSP → Alice: <OCSP Response OK|KO>_{OCSP}
 - Protocol Pros
 - Lighter and simpler than CRL protocol
 - Effective if the adversary is not a MIM
 - Protocol Cons
 - In the clear => confidentiality
 - Exposed to replay attack (nonces are an extension ☹)
 - Browsers silently ignore OCSP if the query times out (=>MIM)

28/04/2021

Public Key Infrastructures

42

There is wide support for OCSP amongst most major browsers:

- Internet Explorer is built on the CryptoAPI of Windows OS and thus starting with version 7 on Windows Vista (not XP[7]) supports OCSP checking.[8]
- All versions of Mozilla Firefox support OCSP checking. Firefox 3 enables OCSP checking by default.[9]
- Safari on macOS supports OCSP checking. It is enabled by default as of Mac OS X 10.7 (Lion). Prior to that, it has to be manually activated in Keychain preferences.[10]
- Versions of Opera from 8.0[11][12] to the current version support OCSP checking.

Conceptually, CRL and OCSP are very close. If you download from a CRL frequently than you get closer to an OCSP. On the other hand, if you query OCSP seldom, then you are getting closer to a CRL.

However, Google Chrome is an outlier. Google disabled OCSP checks by default in 2012, citing latency and privacy issues[13] and instead uses their own update mechanism to send revoked certificates to the browser. More in details:

- For this and other reasons, Google decided in 2012 [to default Chrome not to check for certificate revocation on non-EV certificates](#). (EV or Extended Validation certificates are more expensive certificates with more stringent buyer verification and browser behavior rules.) Instead, Google uses their Chrome update mechanism to send batches of serial numbers of revoked certificates which it constantly gathers by crawling Certificate Authorities.
- **Google's approach trades off some precision**, in that the CRLSet may be old compared to the contents of the current CRL or the response of an OCSP request, **but in exchange the user gains some performance and reliability**. As a Google spokesperson said to me, "...[W]hen we identify a bad cert, we update our revocation list, and Chrome users are automatically protected. CRLSet updates occur at least daily, which is significantly faster than most OCSP validity periods."

Key establishment

CERTIFICATES

apr. '21

Key establishment

43

An example: X.509



A data structure with several fields

- | | |
|-----------------------------------|--------------------------------------|
| 1. Version | 7. Subject public key information |
| 2. Serial number | 8. Issuer unique identifier (v=2,3) |
| 3. Signature algorithm identifier | 9. Subject unique identifier (v=2,3) |
| 4. Issuer distinguished name | 10. Extensions (v=3) |
| 5. Validity interval | 11. Signature |
| 6. Subject distinguished name | |

X.509 uses the Abstract Syntax Notation, ASN.1, (RFC 1422)

X.509 has been conceived for X.400 mail standard

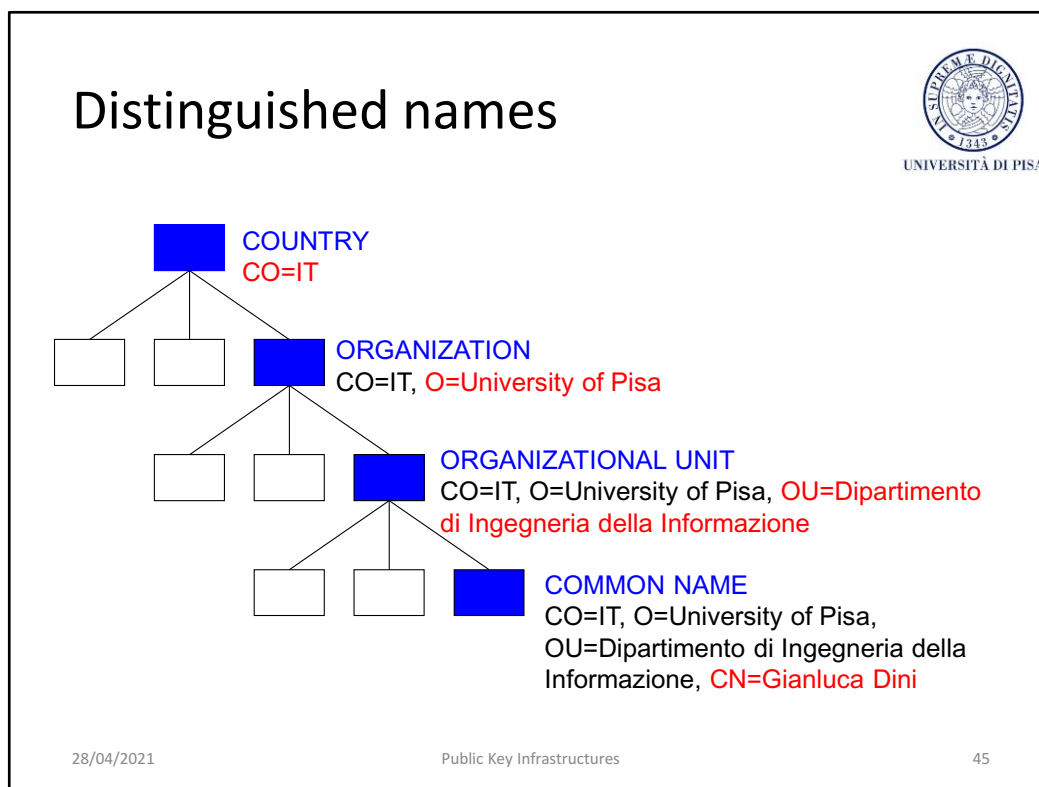
X.509 uses Distinguished Names

28/04/2021

Public Key Infrastructures

44

In the 90' certification was a sort of gold rush. So instead of interoperability, companies (CAs) wanted to lock up their customers by issuing certificates that were not understood by anybody else. For this reason X.509 got **extensions**. Every vendor had its own extensions.



When certificates were invented, certificates and PKIs were implemented by Telco which had (has) a hierarchical view of the world.

A DN should identify a person *uniquely*.

A DN is composed of several fields. Often fields represent elements of a hierarchy.

The hierarchical approach gives us a good way of assigning certificate responsibilities and of ensuring uniqueness. We can assign certification authorities to be responsible for different suffixes. For example there could be a University of Pisa Authority that is responsible of all University of Pisa departments and a separate one for each department. Each authority should guarantee uniqueness among their own certificates. Since each authority issues certificates with a different department and name, no certificates will be duplicated if each authority avoids local duplicates.

Example: <https://www.mps.it>



UNIVERSITÀ DI PISA

Certificate name

www.mps.it

Consorzio Operativo Gruppo MPS

Terms of use at www.verisign.com/rpa (c)00

Florence

Italy, IT

CA 6

Issuer

VeriSign Trust Network

www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign

CA 4 → CA

Details

Certificate version: 3

Serial number: 0x652D0F8ADAB4C7B168A27BBD1C3E9D9D

Not valid before: Mar 2 00:00:00 2004 GMT

Not valid after: Mar 2 23:59:59 2005 GMT

Fingerprint: (MD5) CA CA 88 08 EC D0 8E 49 A6 9A 66 C4 69 31 E0 AE


Fingerprint: (SHA-1) 82 64 CB 69 F0 43 86 43 FF B4 55 D4 25 EF 51 60 65 46 D3 87

contd

28/04/2021

Public Key Infrastructures

46



 UNIVERSITÀ DI PISA

Example: https://www.mps.it

Public key algorithm: rsaEncryption

Public-Key (1024 bit):

Modulus:

```

00: E1 80 74 5E E7 E5 54 8B DF 6D 00 95 B5 96 27 AC
10: 66 93 E0 49 B9 6F 5B 73 53 1C BE 1C EB 47 64 B2
20: 12 95 70 E6 CD 50 67 02 88 E3 EE 9D B1 91 49 C8
30: 8D 58 19 4B 86 8F C0 2E 65 E8 F2 D4 82 CC 55 DB
40: 43 BC 66 DA 44 2F 53 B3 48 4B 37 15 F3 AB 67 C1
50: 69 B4 53 23 19 30 1A 19 23 7F 28 E0 E3 C0 6B 18
60: FF 84 C4 AC A9 74 28 DB FF E9 48 CA 75 D5 35 D6
70: 46 FB 7D D4 A7 3F A1 4B 00 60 14 DC D5 00 CF C7
  
```

SUBJECT'S
PUBLIC KEY

M

Exponent:

```

01 00 01
  
```

$e = 2^{16} + 1$

Public key algorithm: sha1WithRSAEncryption

```

00: 23 A6 FE 90 E3 D9 BB 30 69 CF 43 2C FD 4B CF 67
10: D7 3C 46 22 9A 08 DB 05 1D 45 DC 07 F3 1E 4D 1F
20: 4B 11 23 5B 42 91 14 95 25 88 1F BD 60 E5 6F 84
30: 44 70 7A 95 EC 30 E4 46 4F 37 87 F1 B2 FA 45 04
40: 6F 7C BE 97 25 C7 20 E7 F3 90 55 51 99 3A 72 35
50: 40 F2 E8 E3 36 3A 7D 58 61 9C 91 D6 AC 34 E7 E8
60: 09 27 64 4F 2C 4C C2 D2 A3 32 DB 2B 7E F0 B6 F3
70: 69 96 E4 2B C3 2B 42 ED CA 2C 3C C8 F5 AA E6 71
  
```

CA DIGITAL
SIGNATURE

28/04/2021

Public Key Infrastructures

contd

47

- **Public key algorithm: rsaEncryption** specifies that the certified key is and RSA key
- **Public key algorithm: sha1WithRSAEncryption** specifies that the digital signature algorithm is RSA with SHA-1

Example: <https://www.mps.it>

QUESTO SERVE PER UNA CERTIFICAZIONE AUTHORITY

UNIVERSITÀ DI PISA

Extensions:

- X509v3 Basic Constraints: CA:FALSE
- X509v3 Key Usage: Digital Signature, Key Encipherment
- X509v3 CRL Distribution Points:
 - URI: <http://crl.verisign.com/Class3InternationalServer.crl>
- X509v3 Certificate Policies:
 - Policy: 2.16.840.1.113733.1.7.23.3
 - CPS: <https://www.verisign.com/rpa>
- X509v3 Extended Key Usage: Netscape Server Gated Crypto, Microsoft Server Gated Crypto, TLS Web Server Authentication, TLS Web Client Authentication
- Authority Information Access:
 - OCSP - URI: <http://ocsp.verisign.com>
- Unknown extension object ID 1 3 6 1 5 5 7 1 12:
 - 0_.][0Y0W0U..image/gif0!0.0...+.....k...j.H.,{..0%.#<http://logo.verisign.com/vslogo.gif>

→ POTREI SOSTITUIRE CON UN URL

VA PROTETTO CON LA FIRMA DEL CA

28/04/2021

Public Key Infrastructures

48

Key usage specifica lo scopo della chiave del soggetto. Nel caso in esame

- **Digital Signature** specifica che la chiave può essere utilizzata per verificare firme digitali che supportano servizi di sicurezza diversi dalla firma di un certificato o di una CRL. Ad esempio firme digitali per prova di provenienza e di integrità.

Si noti che questa è diversa da **Non Repudiation** che invece specifica che la chiave può essere utilizzata per servizi di non-ripudio escluso la firma di certificati o di CRL.

- **Key Encipherment** specifica che la chiave può essere utilizzata per il trasporto di chiavi.

Extended key usage specifica uno o più scopi per cui la chiave certificata può essere utilizzata in aggiunta o in sostituzione a quelli specificati da *key usage*.

Key usage ed *extended key usage* devono essere mutuamente consistenti.

Nell'esempio:

- **TLS Web Server/Client Authentication** specifica che la chiave può essere utilizzata per l'autenticazione TLS di un WWW server/client. Questa estensione è compatibile con *digital signature* e *key encipherment*.
- **Microsoft Server Gated Crypto / Netscape SGC** specifica che la chiave può essere utilizzata per scopi privatamente definiti da Microsoft o Netscape.

Basic constraints specifica se il *subject* è una CA oppure no e la lunghezza del path di certificazione.

Nel caso in esame, l'estensione *basic constraints* specifica che il soggetto non è una CA.

CRL Distribution Points specifica come la CRL può essere ottenuta.

Public Key Infrastructure

TRUST MODELS

28/04/2021

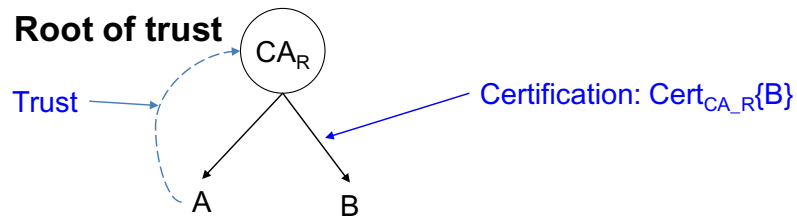
Public Key Infrastructures

49

Centralized Trust Model



UNIVERSITÀ DI PISA



The Model

- Every user trusts the root
- The root releases certificates

Inconvenient

- Users have to go to the root in order to get a certificate

28/04/2021

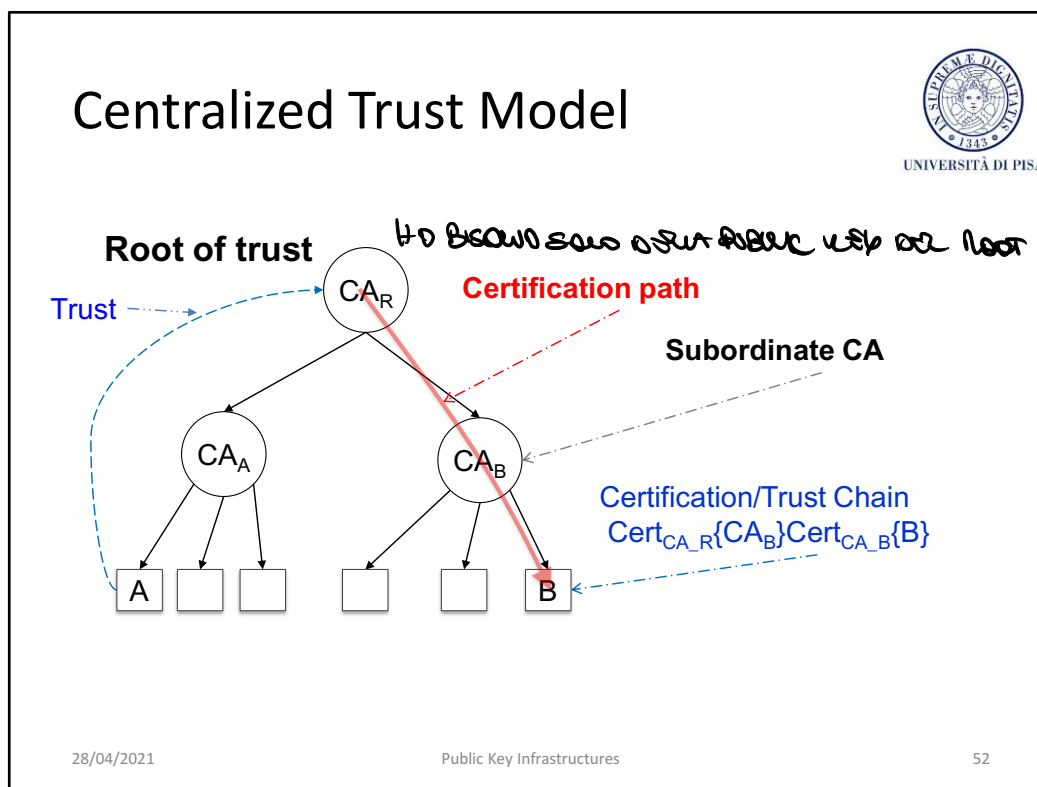
Public Key Infrastructures

51

Telco's and Finance's View

The initial model was very simple.

Root in Geneva, **CCIT** (*Comité Consultatif International Téléphonique et Télégraphique*) now ITU (International Telecommunications Union)



Telco's and Finance's View - The initial model was very simple. Root in Geneve, **CCIT** (*Comité Consultatif International Téléphonique et Télégraphique*) now **ITU** (International Telecommunications Union) certifies **telcos** in every country which, in turn, certify companies, which in turn certify users. ITU specifies the policy.

Why should you trust your telco? Because they are good? Are they really good? Yes! Why? Because they are good.

It's the root which states policies.

However, there are other subjects that are powerful, not only telcos, for example: banks (Visa, Mastercard), companies (Google, Microsoft)

We have this model today. Visa and MasterCard certifies banks which certify customers public keys. If you have a MasterCard/visa credit card with pki enabled you have a card with mastercard/visa logo. Banks are liable for the registration part, i.e. if they give a certificate to the wrong person. However, it is Mastercard/Visa which state the rules (audit procedures, certificate format, etc etc). If you are an MPS customer and go to Australia, your credit card is accepted, even though they don't know MPS, because it carries the Visa/MasterCard logo.

Constrains on the certification path



UNIVERSITÀ DI PISA

- If CA_x certifies CA_y , the trust that CA_x has in CA_y transitively propagates to all CAs reachable from CA_y
- CA_x may limit this propagation by posing constraints
 - **Constraint on the chain length** – The chain after CA_y has a limited length
 - **Constraint on the set of domains** – CAs in the chain after CA_y must belong to a predefined set of CAs

28/04/2021

Public Key Infrastructures

54

Esempio: <https://www.mps.it>



UNIVERSITÀ DI PISA

Certificate name

VeriSign Trust Network

www.verisign.com/CPS Incorporated by Ref. LIABILITY LTD.(c)97 VeriSign

Issuer

VeriSign, Inc.

Class 3 Public Primary Certification Authority

US

Details

Certificate version: 3

Serial number: 0x254B8A853842CCE358F8C5DDAE226EA4

Not valid before: Apr 17 00:00:00 1997 GMT

Not valid after: Oct 24 23:59:59 2011 GMT

Fingerprint: (MD5) BC 0A 51 FA C0 F4 7F DC 62 1C D8 E1 15 43 4E CC

Fingerprint: (SHA-1) C2 F0 08 7D 01 E6 86 05 3A 4D 63 3E 7E 70 D4 EF 65 C2 CC 4F

28/04/2021

Public Key Infrastructures

55

Esempio: <https://www.mps.it>

Public key algorithm: **rsaEncryption**

Public-Key (1024 bit):

Modulus:

00: 6F 7B B2 04 AB E7 34 4F 9C 53 A7 02 B2 90 4F 22
10: F9 3A 3C 5A 8B 51 2B FE CB 42 95 30 70 FE 8A B2
20: D3 1D C1 B8 5A 49 5C F7 39 4E 4D B7 F3 3B 09 F1
30: FA E5 28 93 3E 30 F5 63 AA 43 71 27 56 FE A3 BB
40: CA C4 6C 75 B2 32 C1 07 D9 DD 25 40 F5 5C A9 D4
50: 15 0A 34 9A ED 42 97 EA BD F1 B2 55 45 73 3C AA
60: E7 B6 5B 6C 4C F0 AA 3B 36 E6 BC D3 05 D4 BF E1
70: 2B 65 A2 25 39 18 85 1F 7D 02 19 D6 E8 80 82 D8

Exponent:

01 00 01

Public key algorithm: **sha1WithRSAEncryption**

00: 08 01 EC E4 68 94 03 42 F1 73 F1 23 A2 3A DE E9
10: F1 DA C6 54 C4 23 3E 86 EA CF 6A 3A 33 AB EA 9C
20: 04 14 07 36 06 0B F9 88 6F D5 13 EE 29 2B C3 E4
30: 72 8D 44 ED D1 AC 20 09 2D E1 F6 E1 19 05 38 B0
40: 3D 0F 9F 7F F8 9E 02 DC 86 02 86 61 4E 26 5F 5E
50: 9F 92 1E 0C 24 A4 F5 D0 70 13 CF 26 C3 43 3D 49
60: 1D 9E 82 2E 52 5F BC 3E C6 66 29 01 8E 4E 92 2C
70: BC 46 75 03 82 AC 73 E9 D9 7E 0B 67 EF 54 52 1A


28/04/2021

Public Key Infrastructures

56



Esempio: <https://www.mps.it>


 UNIVERSITÀ DI PISA

Extensions:

- X509v3 **Basic Constraints**: CA:TRUE, pathlen:0
- X509v3 **Certificate Policies**:
 Policy: 2.16.840.1.113733.1.7.1.1
 CPS: <https://www.verisign.com/CPS>
- X509v3 **Extended Key Usage**: TLS Web Server Authentication, TLS Web Client Authentication, Netscape Server Gated Crypto, 2.16.840.1.113733.1.8.1
- X509v3 **Key Usage**: Certificate Sign, CRL Sign
- Netscape Cert Type: SSL CA, S/MIME CA
- X509v3 **CRL Distribution Points**:
 URL: <http://crl.verisign.com/pca3.crl>

Certification Practice Statement

Handwritten notes and diagrams:

- VS (Verisign) in a circle
- VS TJ (Verisign Trusted Jurisdiction) in a circle
- VS TJ 2 (Verisign Trusted Jurisdiction 2) in a circle
- Red text: ~~NON AMMESSO!~~ AUTA SUBORDINATO CA
- Green text: AMMESSO
- VS Root CA in a circle
- VS TJ SUBORDINATO CA in a circle
- VS TJ CUSTOMER in a circle
- 28/04/2021
- Public Key Infrastructures
- 57

Key usage specifica lo scopo della chiave del soggetto. Ad esempio

- **Certificate Sign** specifica che la chiave può essere utilizzata per verificare firme digitali su certificati.
- **CRL Sign** specifica che la chiave può essere utilizzata per verificare firme digitali su una CRL.

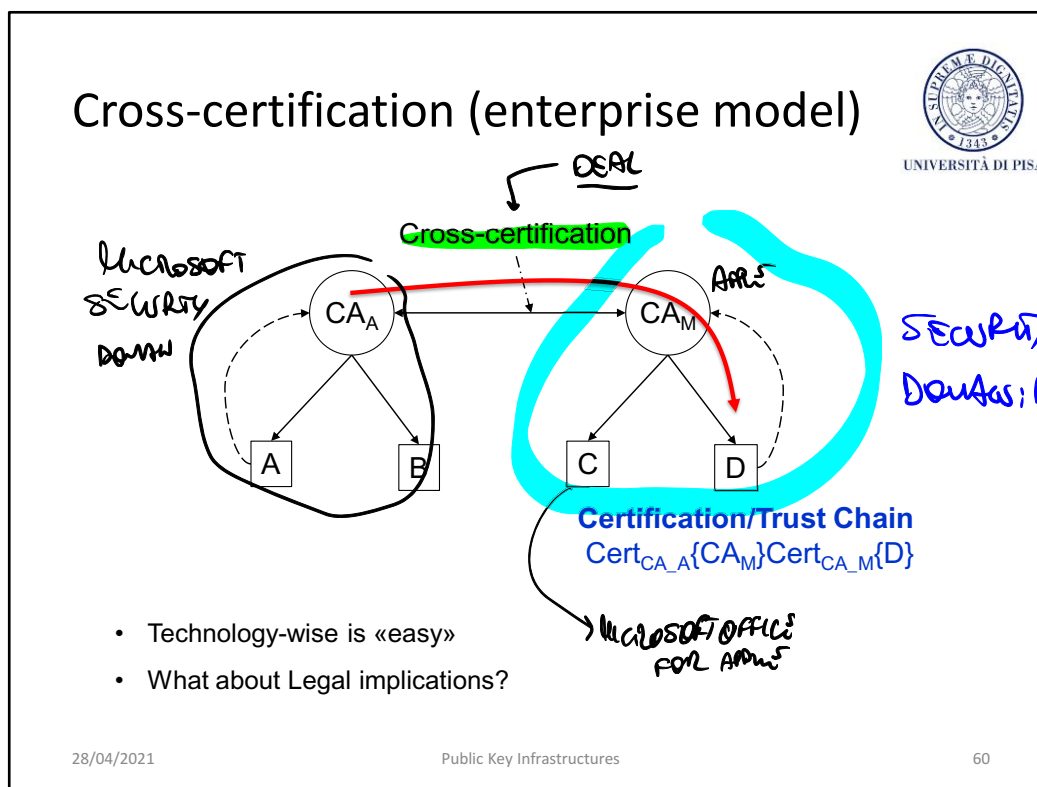
Basic constraints specifica se il *subject* è una CA oppure no e la lunghezza del path di certificazione.

Nel caso in esame, l'estensione *basic constraints* specifica che il soggetto è una CA e che sono ammessi percorsi di certificazione lunghi zero. Ciò significa che **questa CA non ne può delegare/certificare altre**.

CPS

The Certification Practice Statement ("CPS") states the practices that CA employs in providing certification services that include, but are not limited to, issuing, managing, revoking, and renewing certificates in accordance with the specific requirements of Certificate Policies ("CP").

The CP establishes the business, legal, and technical requirements for approving, issuing, managing, using, revoking, and renewing, digital Certificates and providing associated trust services.



This model is appropriate for companies.

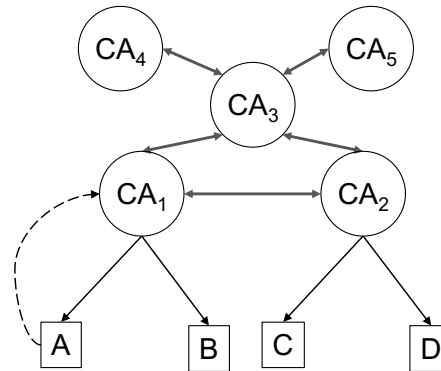
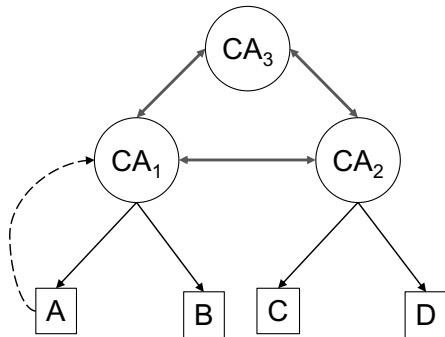
For example, when Apple decided to put Office on the iPad, then Apple and Microsoft made a deal according to which authenticated updates of Office must be accepted by Mac OS X. Similarly, an application to access the Apple Cloud runs over Windows. This requires a cross-certification. Technology-wise is simple. **What about legal implications?**

Enterprise model



UNIVERSITÀ DI PISA

Hub-and-spoke



Business requirements make certification complex

- No hierarchy anymore but mutual agreement
- Customers trust local CAs

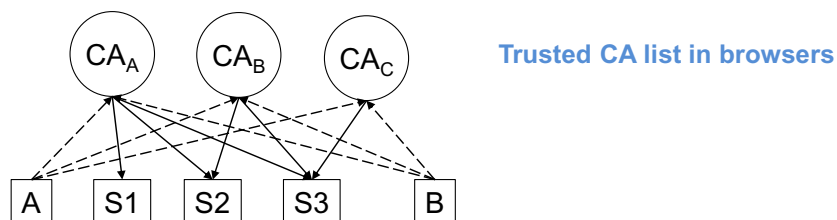
28/04/2021

Public Key Infrastructures

61

Generalization of the cross-certification template

Browser model



- More levels are possible
 - Subordinate CAs
- A user trusts **all** CAs in his browser
 - There are 650 CAs but many of them are related => 75

28/04/2021

Public Key Infrastructures

62

When you establish an SSL/TLS connection with a server, the connection succeeds if the server has been certified by one of these CAs. Which one? You can't know. You have to trust all the CAs.

This list even contains the Turkish and the Chinese Government! You aren't expected to verify by hand who vouched for whom in the hierarchy of trust. It all happens automatically when your browser sets up a secure connection. Root CAs are the starting point of the three of trust. Their certificates are pre-loaded in your browser and automatically bestow trust downwards. It is as if Microsoft, Mozilla, Apple vouched for a root CA.

In order to get into that list you just need money. The system is designed to make money not security. Paradoxically, if you open a CA and don't charge for a certificate then they will stop you.

The TURKTRUST case

- "By mistake", Turktrust, which is in the trusted CA list, released a subordinate CA cert to EGO (Turkish authority for public transportation). So EGO had the ability to sign certificates to any domain name it chose.
- EGO decided to implement *security scanning of HTTP traffic*.
 - HTTP security scanning is easy => proxy
 - HTTPS security scanning is hard => you have to mount a MIM attack, i.e., split an HTTPS connection in two connections: *browser2proxy* and *proxy2server*. This is also called *keybridging* or *decrypt-recrypt* but ultimately is a MIM. It is not an attack as long as it is your company's outbound traffic. The pain is that the user gets continuously warnings because the SSL connection terminates at the proxy and not at the intended server. A way to solve this problem is the following:
 - Build your own CA. Install the CA on the proxy and put the CA-cert in the browser.
 - Let the proxy CA generate a placeholder certificate *pCert = SvrlId, ProxyPubK, ...,signature*
 - BYOD and contractors get warnings until they don't install the proxy CA cert.
- EGO performed keybindings putting TURKTRUST/EGO in the proxy .
- A few days later a Chrome user received a warning from google about an unexpected certificate claiming to represent a Google web property.
 - This is because Chrome implements *public key pinning*: the browser is equipped with a list of presumed-good CAs but also with a list of known-good Google SSL certs. So even though a presumed-good CA suddenly starts signing certs for *.google.com the browser will complain. This helps to protect against the compromise of a root CA or deliberate dodgy behaviour.

The CA Mess on the Web



- **Recommended reading**

- An Observatory for the SSLiverse, Peter Eckersley, Jesse Burns, [Defcon 18](#), Las Vegas, USA, July, 2010 ([pdf](#), [video](#))

Incidents



- March 2011 – Comodo
 - 9 fraudulent certs
- Summer 2011 – DigiNotar
 - 500+ fraudulent certs
 - [FOX-IT final report \(long\)](#)
 - [ENISA's resume \(short\)](#)
- January 2013 – Turktrust
 - 100+ fraudulent certs

28/04/2021

Public Key Infrastructures

64

In all these cases, false certs for high profile websites such as Google, Yahoo, Mozilla etc etc issued made. This makes it possible to perform man-in-the-middle attacks.

In the cases of COMODO and DigiNotar, CAs suffered from a compromise. Actually, in the case of COMODO a subordinate CA of COMODO's was compromised. This reminds us that certification is just trust delegation:

When a CA provides its partner with the ability to issue certificates in its name, they were not only trusting said partner to properly vet the identities for certificate applicants, but also trusting the partner to protect this power to issue certificates.

In the case of TURKTRUST instead, CA said they suffered from a misconfiguration which reported themselves.

CA incidents and compromise may lead to MIM attacks. But, are incidents really incidents? Especially when a gov is involved.

Countermeasures

- Prompt detection of compromise/misconfiguration
- Revocation by CRL and/or OCSP
- Activate browser's certificate revocation option => *how does your browser manage revocation?*
What is going to do when it doesn't find the CRL/OCSP service (no answer)?

Countermeasures (intuitions)



- Public key pinning
 - List of presumed-good CAs and list of known-good certs
 - Chrome
- Certificate transparency
 - To make public that a CA issued a cert
 - Resistance from business
- Convergence
 - Download a cert directly and from a set of trusted CAs and compare them
- DANE (DNS-based Authentication of Name Entities)
 - Store a pubK in a DNS record; require DNSSEC
- Extended Validation certificates
 - Prove the legal entity controlling the website or sw package...
 - ...promise what we were promised a decade ago and we never got ([The inevitable collapse of the certification model](#), Hagai Bar-EI)

28/04/2021

Public Key Infrastructures

65

According to Hagai Bar-EI, the certification model failed because of the economics behind it:

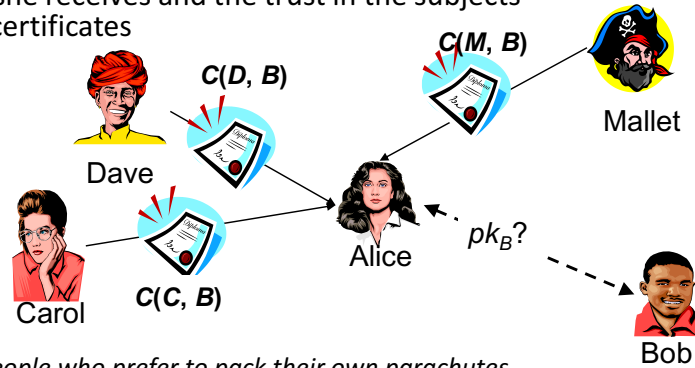
1. **Race to the bottom** – Once the CA gets the money from the certificate applicant, the less effort it spends on the issuance process, the higher profit it makes. Sophisticated scrutiny of the entity seeking certification costs money, and as scrutiny can also lead to rejection — it leads to fewer customers as well. Giving away certificates after milder checks makes better margins, and brings more business.
2. **Externality** - If the damage of lax certification practices was borne by the customers buying the certificates, then we could expect free economy to correct the situation by having such CAs lose customers for CAs with more stringent certification practices. Unfortunately, this is not the case. The damage caused by lax certification practices is borne by the public, not by the certificate buyers. We suffer from nation- and corporate-caused MITM attacks, phishing and impersonation on the net, caused by offenders getting certificates under others' identities.

Personal trust model (PGP model)



UNIVERSITÀ DI PISA

- The user decides how much trust to put in a certificate
 - Alice determines the trust in pk_B according to the number of certificates she receives and the trust in the subjects issuing the certificates



PGP is for people who prefer to pack their own parachutes
[P. Zimmerman]

28/04/2021

Public Key Infrastructures

66

PGP model - Validity and trust level



UNIVERSITÀ DI PISA

- Trust level in a key
 - Own key
 - Implicit trust
 - Others' keys
 - Complete trust
 - Marginal trust
 - No trust
 - A key may be
 - Valid
 - Marginally valid
 - Invalid
 - A key is valid if it has been signed by a completely trusted key or by two marginally trusted keys
- The user defines the trust to put in a key*

28/04/2021

Public Key Infrastructures

67

PGP vs X.509



- Number of signatures
 - X.509 – A key is signed just once
 - PGP – A key may be signed multiple time
- Trust level
 - X.509 – A certificate is implicitly associated to a certain trust level
 - Depend on the CA policy
 - PGP – Every signature is associated to an explicit trust level
 - Signatures on the same key may have different trust levels
 - The meaning of a trust level depend on the context

28/04/2021

Public Key Infrastructures

68

Personal Trust Model – PGP cons



UNIVERSITÀ DI PISA

- Hard to understand if you're not an expert
- Key revocation is a nightmare

28/04/2021

Public Key Infrastructures

69

Browser behaviour



- Idealized model
- Reality
 - Revocation is blocking information (latency)
 - What if revocation infrastructure is unreachable?
 - Browsers have been forced to ignore revocation information when unavailable
 - Types of server certificates
 - DV, OV => not checked by default
 - EV => checked but, if unavailable, response is browser-dep

[Defective By Design? - Certificate Revocation Behavior In Modern Browsers](#), SpiderLabs Blog, Apr. 4, 2011

28/04/2021

Public Key Infrastructures

70

Opera is the most aggressive of the major browsers. The connection will be labeled unencrypted if revocation checks fail.

In-house or external CA?



UNIVERSITÀ DI PISA

- Implement your own CA or exploit a commercial one?
 - Cost-convenience ratio
 - High quality certification \Rightarrow high costs
 - Low quality certification \Rightarrow high risks
 - In-house
 - Pros – Complete control of the certification process
 - Cons – Cost of the infrastructure; limited scale
 - Commercial
 - Pros – Large scale
 - Cons – Trust delegation; no liability

28/04/2021

Public Key Infrastructures

73

