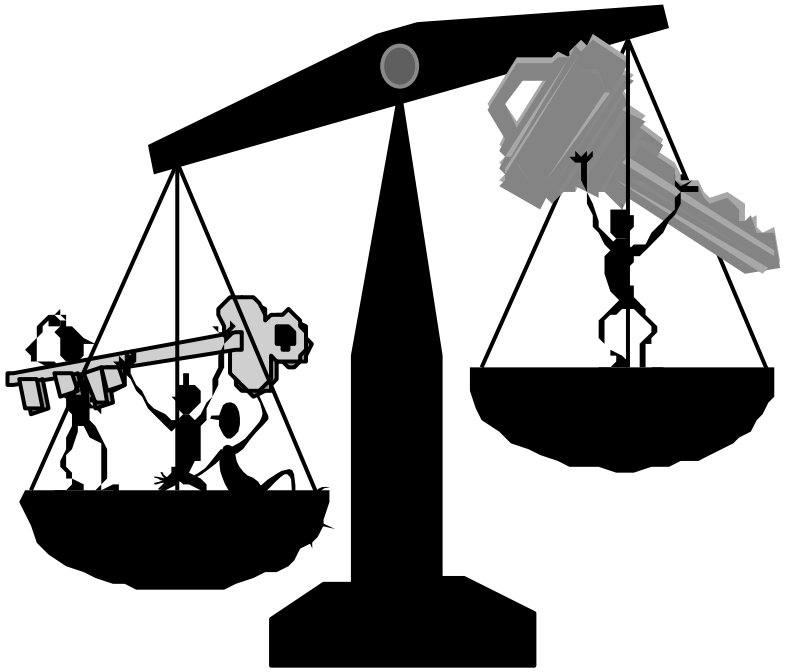
Meccanismi asimmetrici



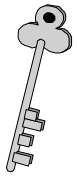
Crittografia a chiave pubblica

- **Cifrari**
- **Generatori di bit pseudocasuali**
- **Schemi di firma**
- **Protocolli d'identificazione attiva**

Ogni utente ha una chiave segreta SU

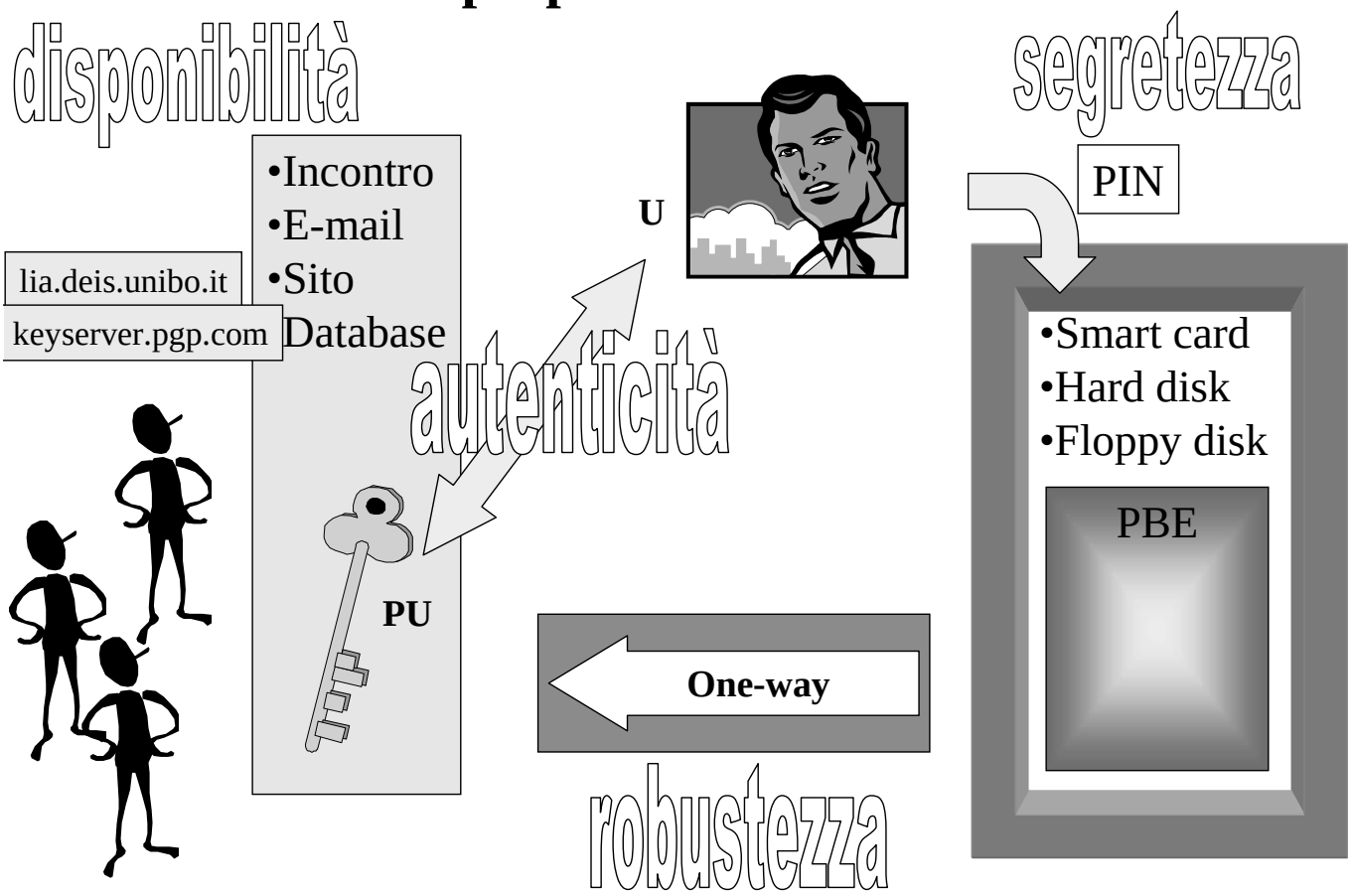
Chiave di firma o di decifrazione

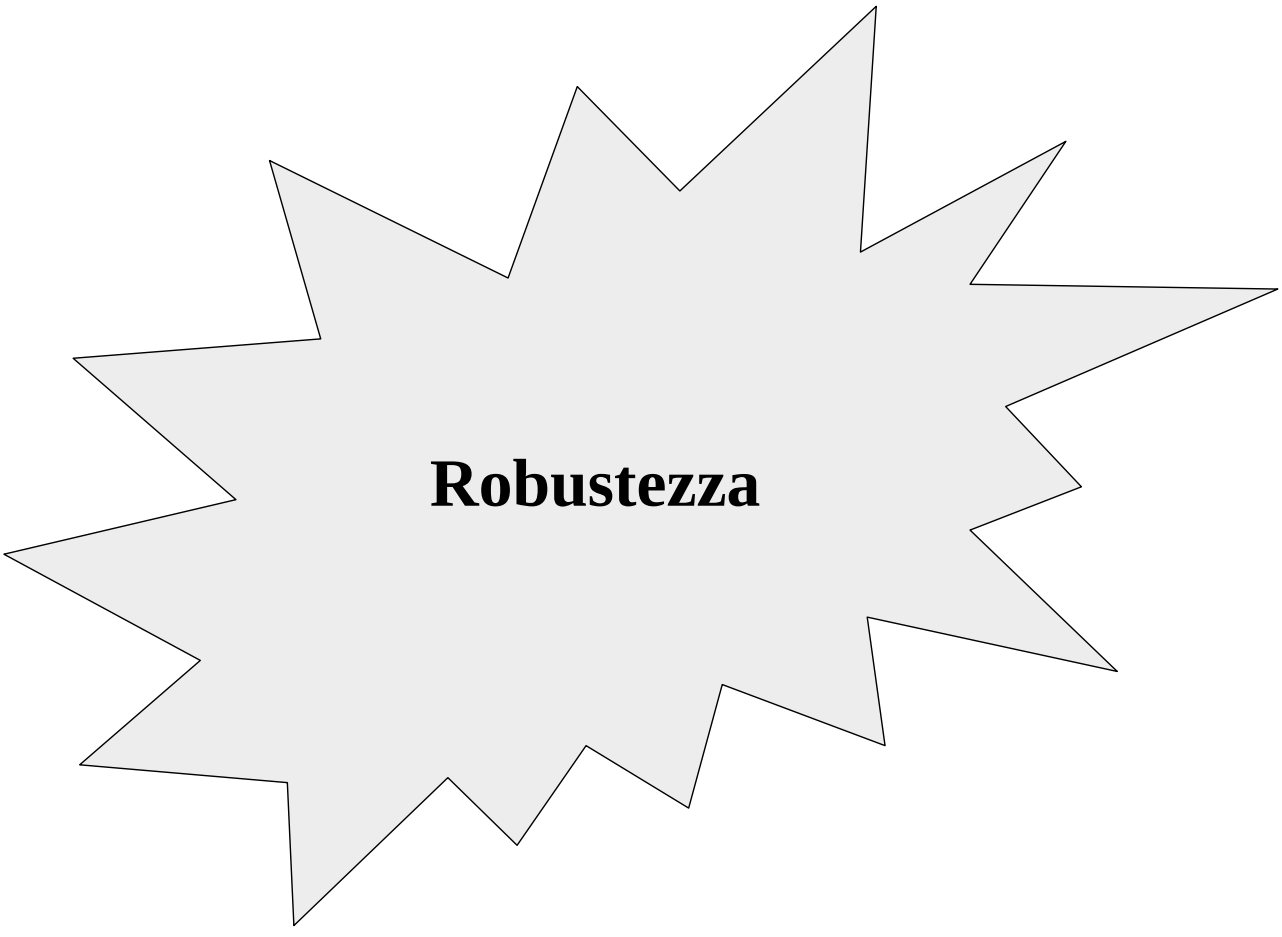
..e rende pubblica una chiave PU



Chiave di verifica o di cifratura

Le proprietà delle chiavi

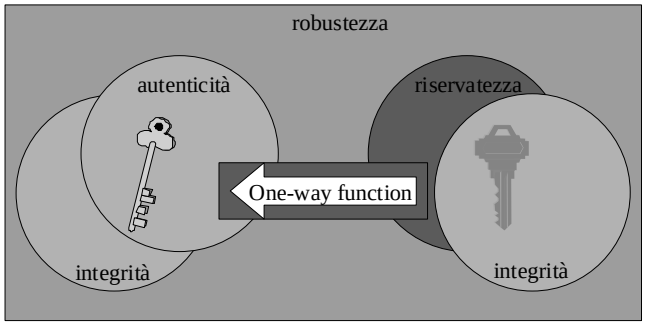




La relazione tra le chiavi

Una funzione **f** è unidirezionale se è invertibile, se il suo calcolo è facile e se per quasi tutti gli **x** appartenenti al dominio di **f** è difficile risolvere per **x** il problema **y = f (x)**.

Proprietà delle chiavi asimmetriche



Chiave pubblica **Chiave privata**

x, y devono inoltre selezionare una coppia di trasformazioni una inversa dell'altra e facili da eseguire:

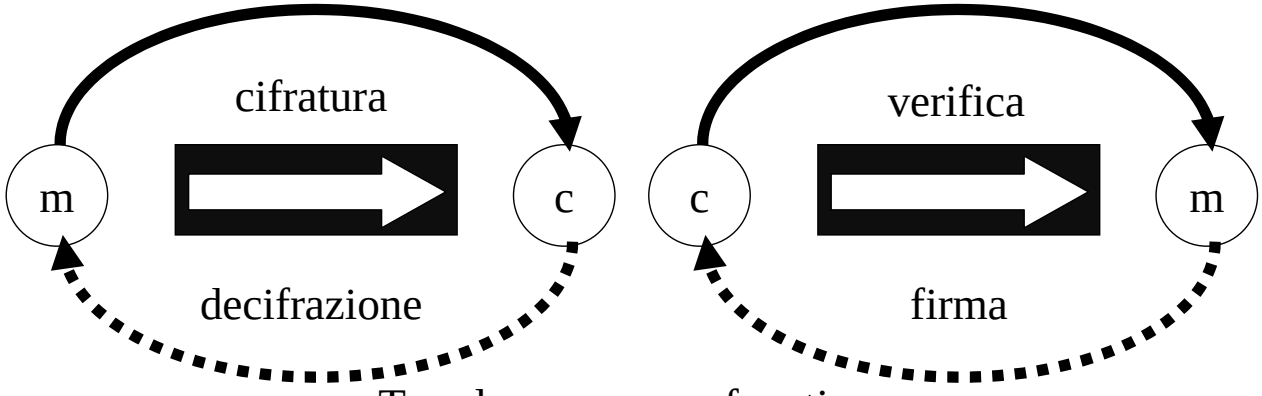
- $E_y \& D_x$ **cifrario**
- $S_x \& V_y$ **firma digitale**

Funzioni "candidate"

Problemi difficili della Teoria dei numeri

Le due trasformazioni

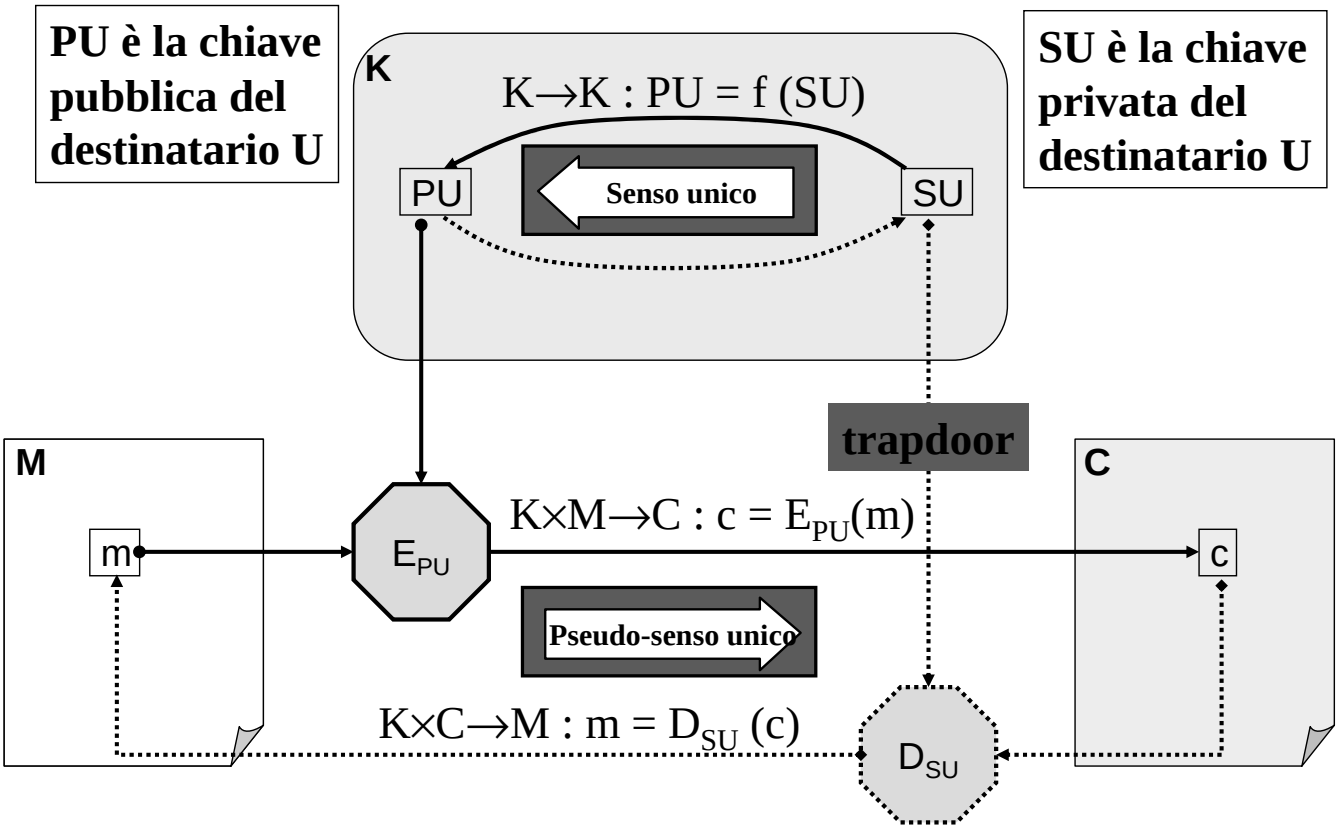
Una funzione **f** è pseudo-unidirezionale se appare come unidirezionale per chiunque non sia in possesso di una particolare informazione sulla sua costruzione (**trapdoor**).



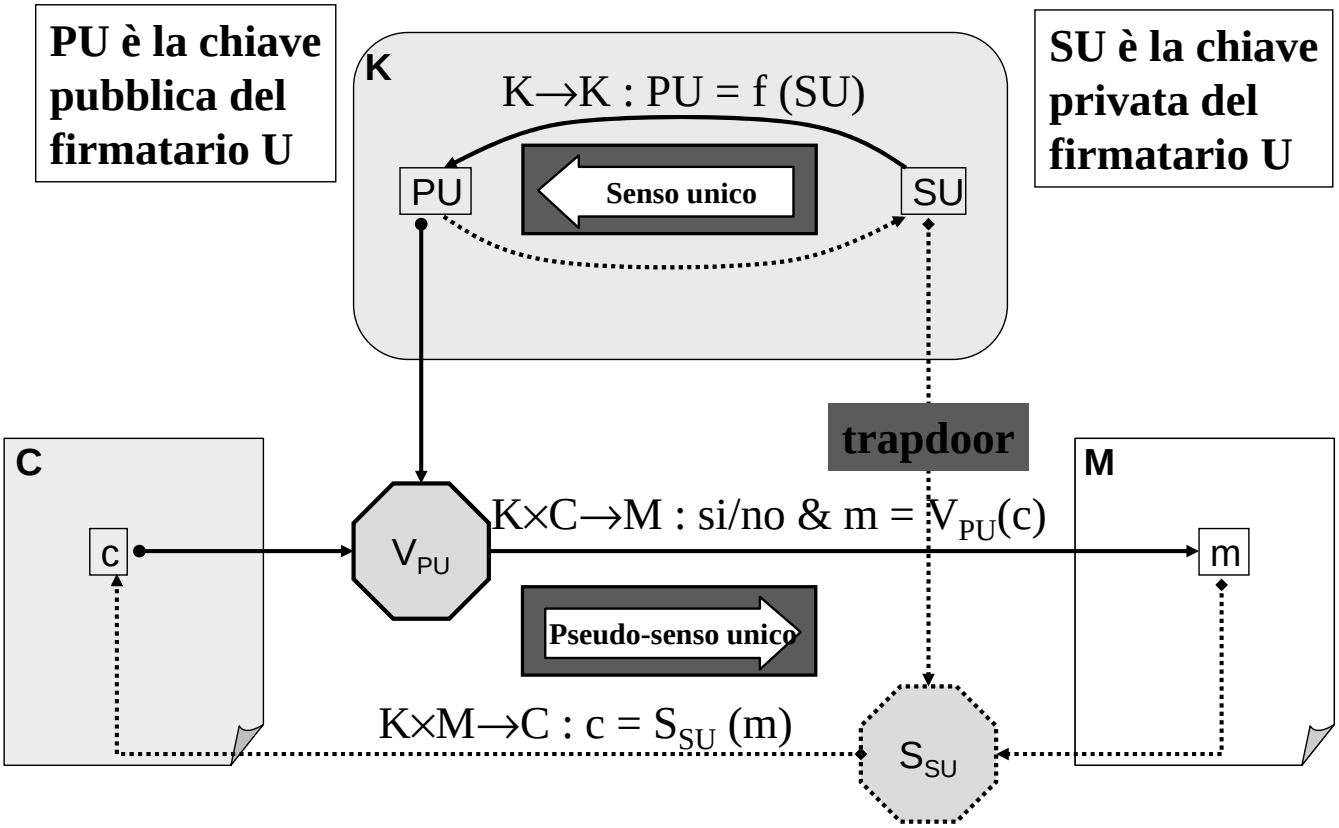
Trapdoor one-way function

Allo stato attuale delle conoscenze la trasformazione segreta degli algoritmi asimmetrici è facile da calcolare solo per chi ha la chiave privata. Probabilmente un altro metodo facile non verrà mai trovato

Il modello del Cifrario a chiave pubblica



Il modello della Firma digitale a chiave pubblica



Problemi difficili

Assunzione: per certi problemi della Teoria dei numeri non si troveranno mai algoritmi con tempo polinomiale

P1: logaritmo discreto (gruppo ciclico o GF(p^n))

Scambio DH, Cifrario ElGamal, Firma DSS
Safe prime: **p = 2q+1** con **q** *primo di Sophie Germain*
q si ottiene dalla progressione 6k-1

P2: radice e-esima - Dato un **n** prodotto di due primi, un **e** coprimo con Φ(n) ed un elemento **c** ∈ $\mathbf{Z_n}$ trovare un intero **m** tale che

$$m \equiv \sqrt[e]{c}$$

Cifrario RSA
Il problema è facile se si conoscono i fattori di n o se n è primo

P3: fattorizzazione

P3: problema della fattorizzazione - Dato un intero positivo **n**, trovare i numeri primi $\mathbf{p_i}$ ed i coefficienti interi $\mathbf{e_i \geq 1}$ tali che
$\mathbf{n = p_1^{e1} \times ... \times p_k^{ek}}$ (Crittografia asimmetrica: $\mathbf{n= p_1 \times p_2}$)

Cifrario RSA

- Gauss e Fermat: **20** cifre decimali
- 1970: **41** cifre decimali con un main frame
- 1977, Rivest: **125** cifre decimali è un calcolo impossibile
- 1994: **129** cifre decimali in 8 mesi con 1.600 workstations
- 2000 (stima): **150** cifre decimali in un anno con una macchina parallela da 10 milioni di dollari
- 2004: TWINCLE, setaccio "optoelettronico" (2-3 ordini di grandezza)
- 2004 (prev.): **300** cifre decimali (1024 bit)
- 2014 (prev.): **450** cifre decimali (1500 bit)

Algoritmi di fattorizzazione

Trial division:
1. i=1
2. individua p(i): i-esimo primo
3. calcola n/p(i)
se ***resto*** **≠ 0**, i = i+1 e goto 2
altrimenti **I° fattore =** ***quoziente***

Rivest, 1999
scelta casuale

I° fattore < sqrt(n)
O(exp (½ (log n)))

. . . . .

"p, q *strong primes* : p-1, p+1, q–1, q+1 con grandi fattori primi

"$|p-q| > n^{1/2}$ "

General Number Field Sieve:
$O(\exp (\log n)^{1/3} . (\log(\log n))^{2/3})$

1024 – 2048 bit

Altri problemi difficili

P4: problema del fusto - Dato un insieme di **n** interi positivi

$\{a_1, a_2 , .., a_n\}$

ed un intero positivo **s** determinare se esiste o meno un sottoinsieme di $\mathbf{a_j}$ la cui somma è **s**.

Cifrario di Merkle-Hellman, Cifrario di Chor-Rivest

P5: problema della radice quadrata modulare - Dato un **n** composto ed un elemento **$a \in Q_n$** (insieme dei residui quadratici modulo n) trovare la sua radice quadrata modulo n, cioè
un intero **$x \in Z^*_n$** tale che **$x^2 \equiv a \pmod n$**.

Cifrario di Rabin

N.B. Il problema è facile se si conoscono i fattori di n o se n è primo

P6: problema della residuosità quadratica - Dato un **n** composto e dispari ed un elemento **$a \in Jp$** (insieme degli interi con simbolo di Jacobi = 1) determinare se **a** è o meno un residuo quadratico modulo **n**.

PRNG BBS, Cifrario di Goldwasser-Micali

Logaritmo discreto su curva ellittica

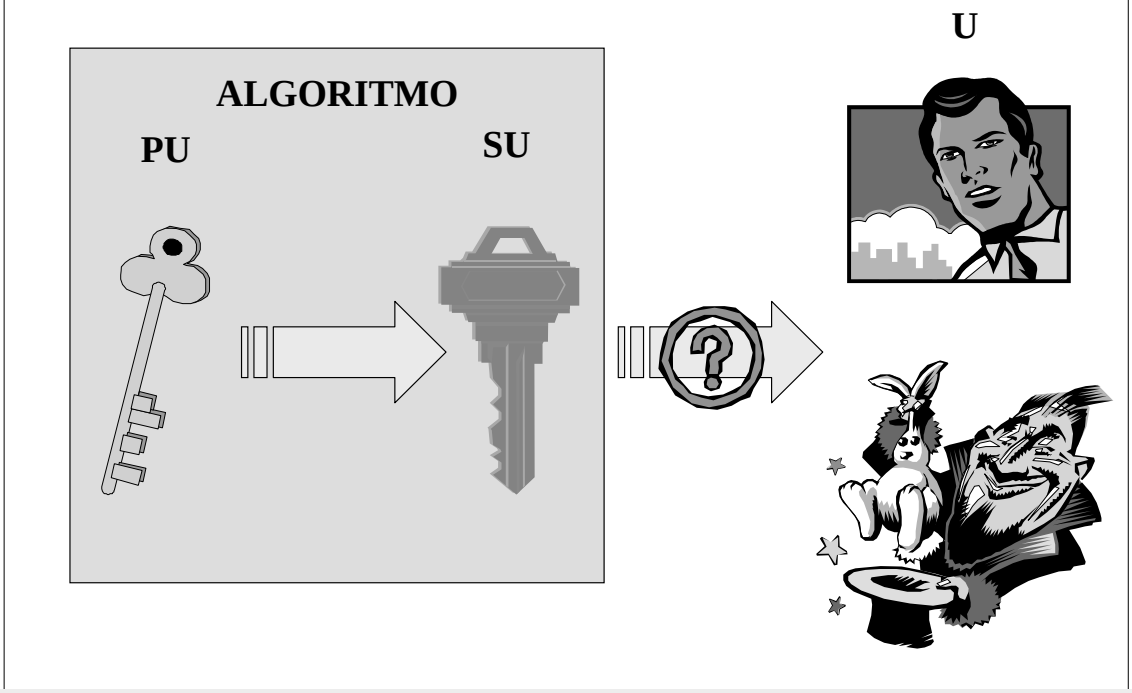
P7: problema del logaritmo discreto su una curva ellittica - Data la curva ellittica formata da punti le cui coordinate x,y soddisfano l'equazione
$y^2 = x^3 + ax + b \mod p$, con p primo,
e due suoi punti **P**, **Q** tali che $Q = n \times P$,
determinare **n**.

Complessità degli attuali algoritmi di rottura
$O(\exp(\frac{1}{2}(\log p)))$
160-180 bit

ECC(Ellipltical Curve Cryptography)
Certicom.com

Autenticità della chiave pubblica
- **l'attacco dell'uomo in mezzo**
- **il certificato**
- **la PKI**

Chiavi e Proprietario

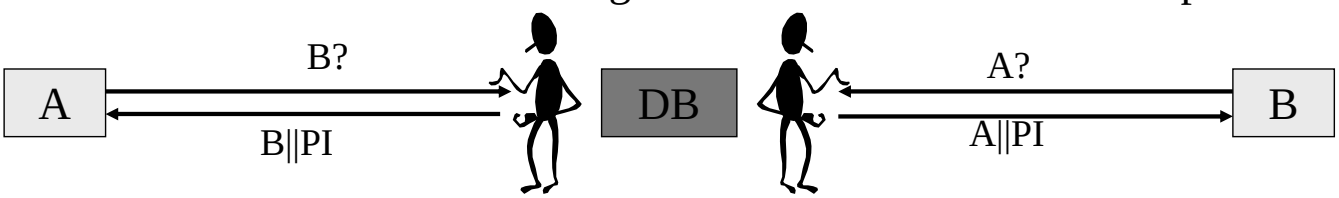


R28: "*prima d'impiegare una chiave pubblica bisogna o essere certi dell'identità del suo proprietario o poterla verificare*"

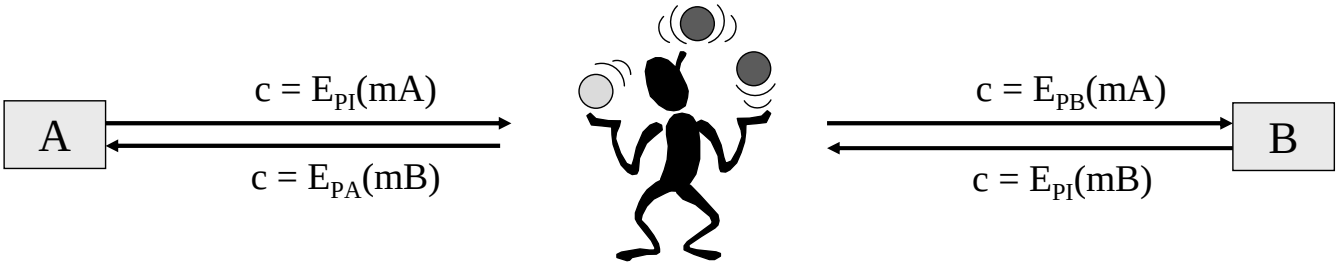
Attacco dell'uomo in mezzo

1 - Registrazione

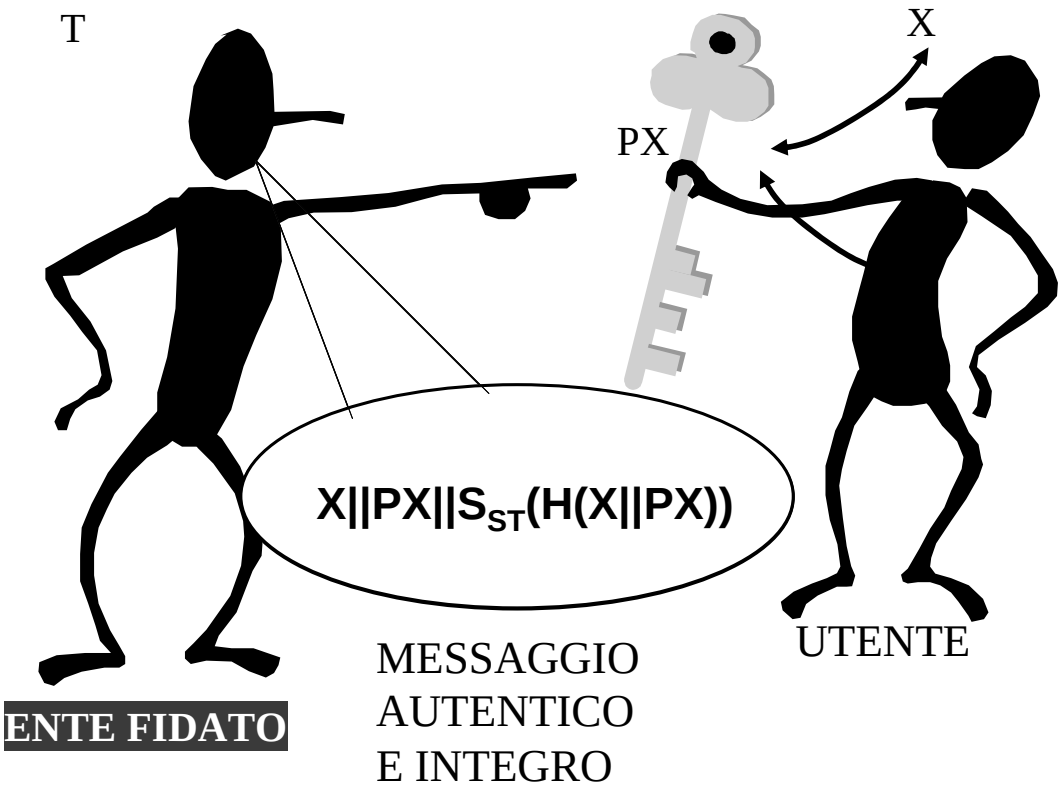
2 - Intercettazione delle interrogazioni e falsificazione delle risposte



3 - Intercettazione, decifrazione, cifratura ed inoltro.



Autenticazione di una chiave pubblica



Preparazione ed uso di un certificato

X identificato in modo sicuro

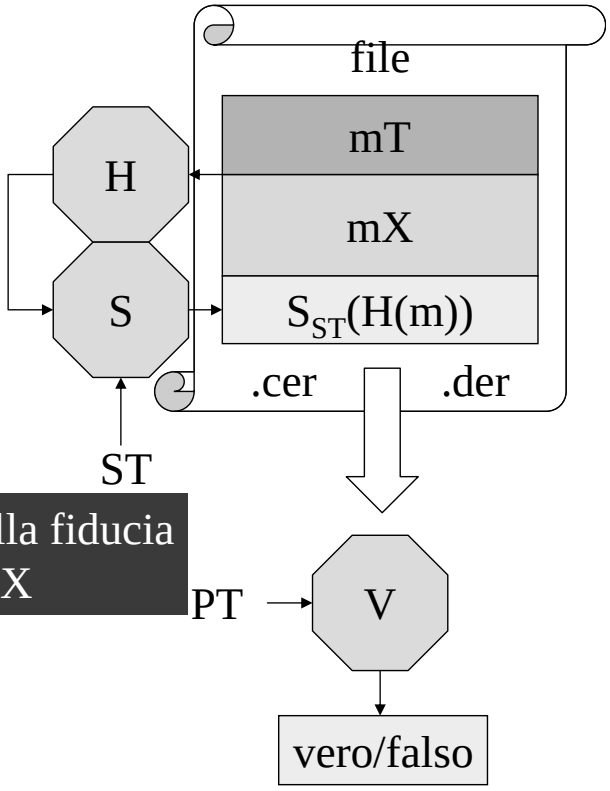
- mX = X||IX||PX||IPX

- mT = T||IT
- m = mT||mX

- H(m)
- $S_{ST}(H(m))$
- Cert(PX,T) = m|| $S_{ST}(H(m))$

PT nota in modo certo

- $V_{PT}(Cert(PX,T))$

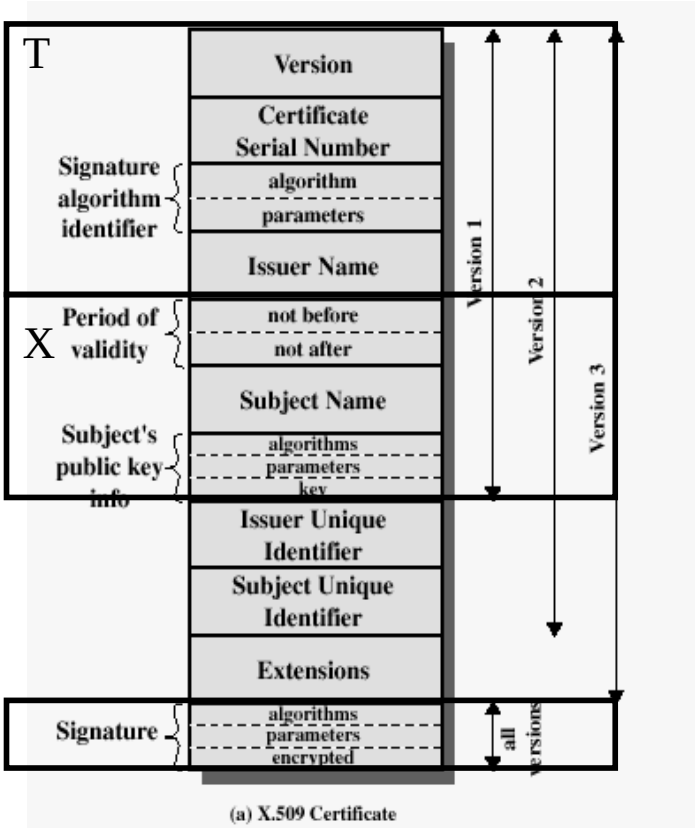


Il Certificatore

T?

- ENTE ufficialmente riconosciuto
  Certification Authority o CA (X.509)
- Qualsiasi UTENTE
  (PGP)

X.509 Formats



(a) X.509 Certificate

Standard Certificate Extensions (1.)

- *version 3 introduces a mechanism whereby certificates can be extended, in a standardized and generic fashion, to include additional information;*
- *certificates are not constrained to only the standard extensions and anyone can register an extension with the appropriate authorities (e.g., ISO);*
- *standard extensions for public key certificates can be separated into the following groups:*
  - *key information;*
  - *policy information;*
  - *user and CA attributes;*
  - *additional information (crl distribution point, ..)*

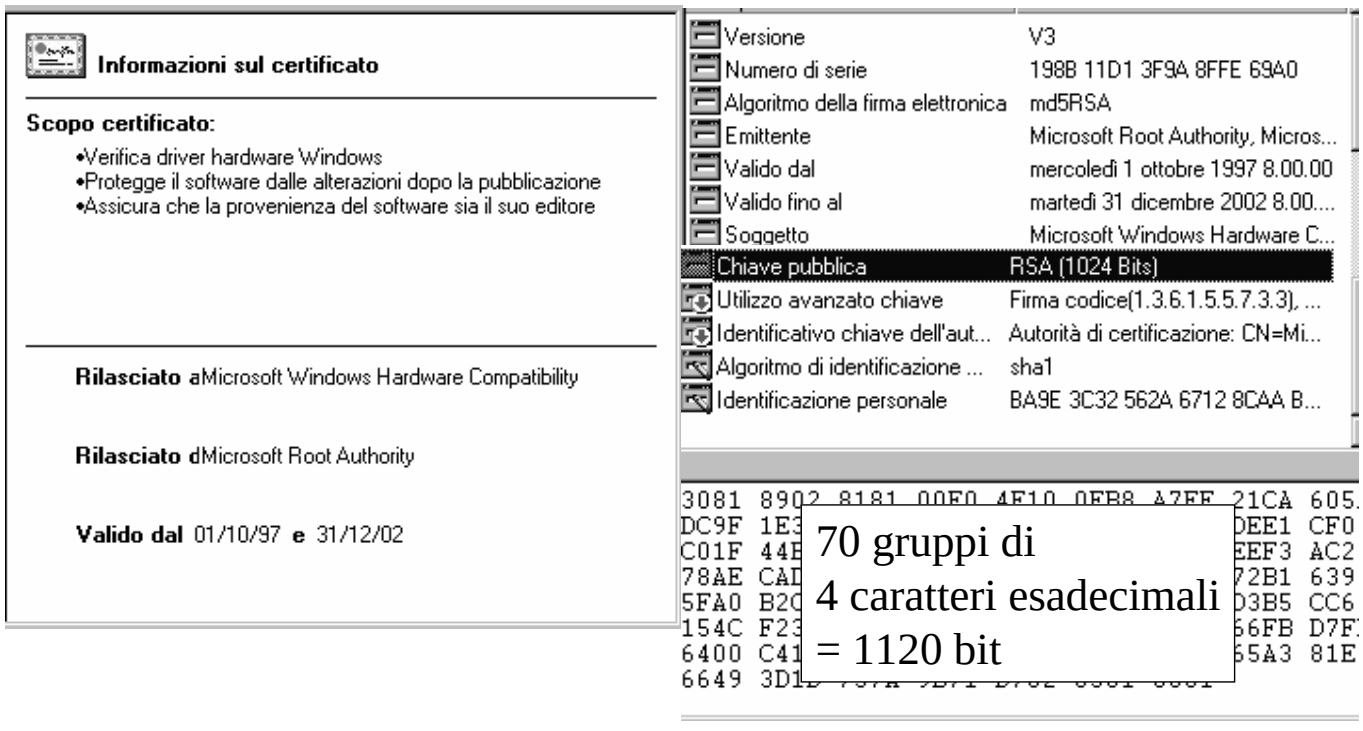
Standard Certificate Extensions (2.)

- ***authority key identifier**: specifies a unique identifier of the key pair used by the CA to sign the certificate;*
- ***subject key identifier**: serves much the same purpose as the authority key identifier;*
- ***key usage**: specifies the intended use(s) of the key. The following list represents the settings for the key usage field: certificate signing (e.g., a CA key pair), CRL signing, digital signature, symmetric key encryption for key transfer, data encryption (other than a symmetric key);*
- ***private key usage period**: specifies the date on which the signing private key expires for a user's digital signature key pair*

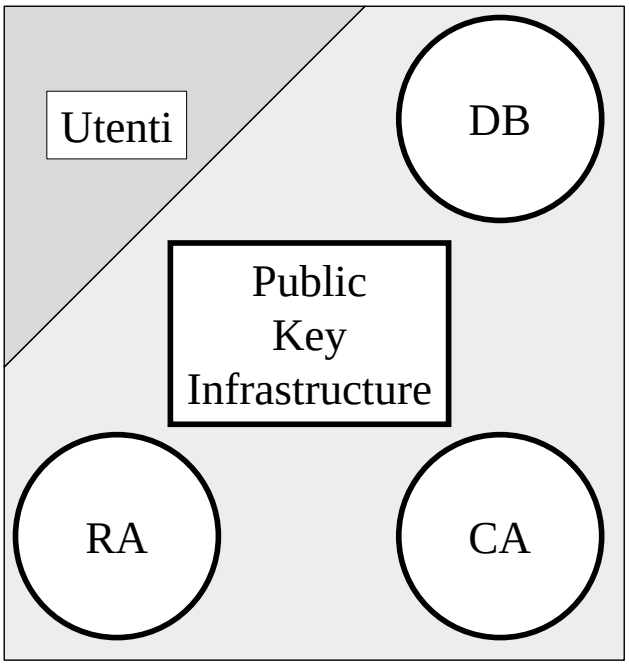
Standard Certificate Extensions (3.)

- ***subject alternative name****: specifies one or more unique names for the certificate subject; the permissible name forms are Internet e-mail address, Internet IP address, , web URL*
- *the policy information extensions provide a mechanism for the CA to distribute information regarding the ways a particular certificate should be used and interpreted;*
- ***certificate policies****: specifies the policies under which the certificate was issued to the user and/or the types of uses applicable to the certificate; certificate policies are represented by specially-formatted numbers, known as object identifiers;*

Explorer: un certificato X.509



PKI: RA, CA e Directory

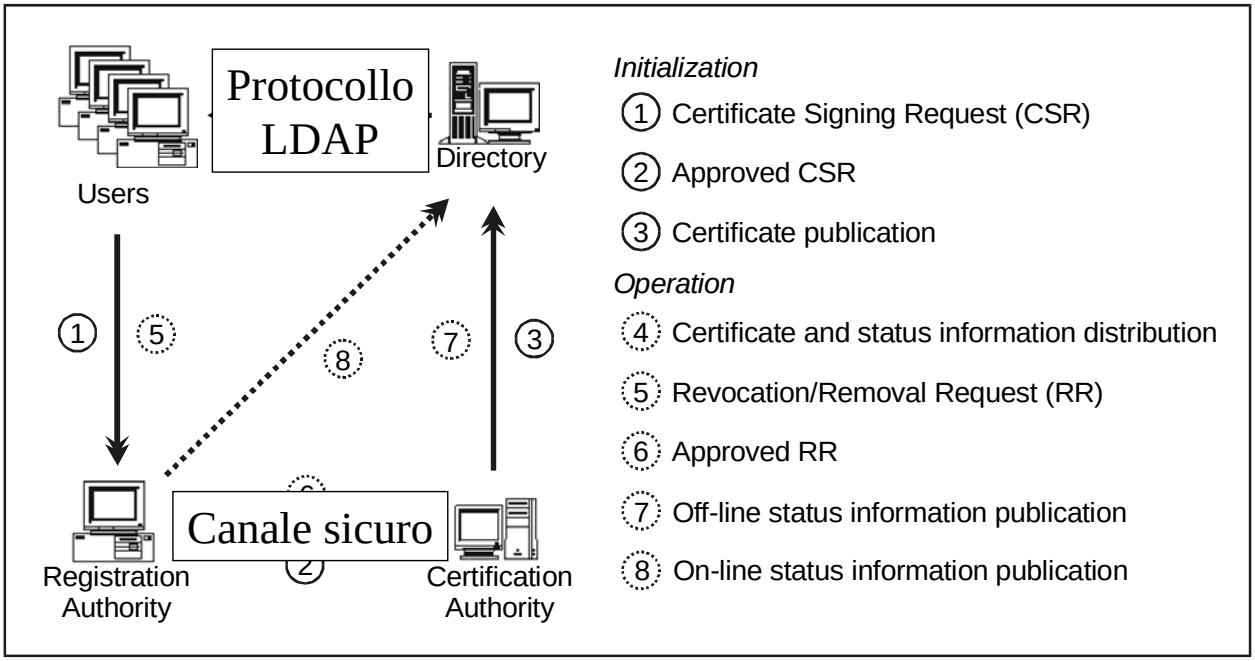


Transazioni

- U → RA
- RA → CA
- CA → U
- CA → DB
- U ↔DB

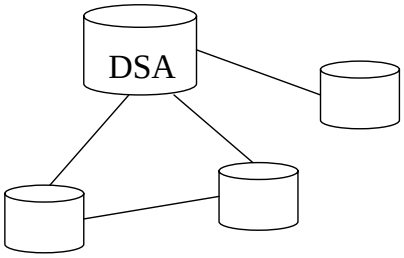
Verisign
Actalis
Trustitalia
E-trustcom
…

PKI: RA, CA e Directory



What is a Directory?

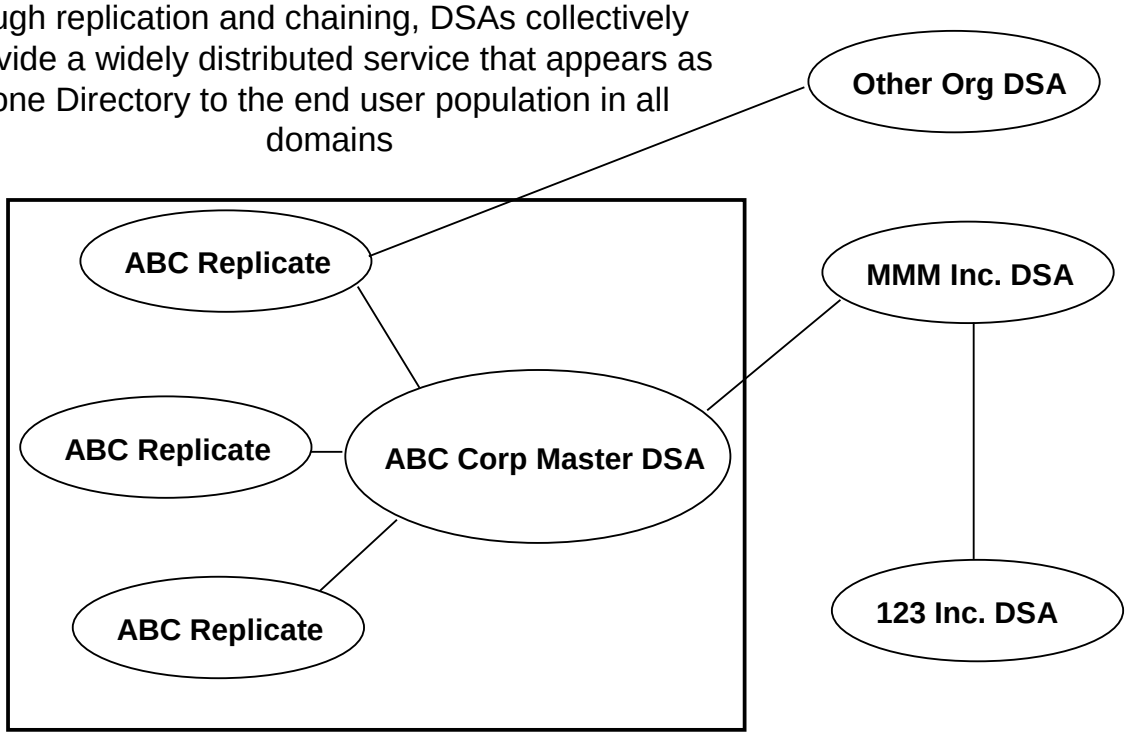
- *Intended as a global repository of information*
  - *Used for:*
    - *electronic mail*
    - *cryptographic exchanges*
    - *telephone systems*
- *In essence the Directory is a distributed database, capable of storing information about people and objects in various nodes or servers distributed across a network*



X.500 - The Directory

DirectoryUserAgents and DirectoryServiceAgents

Through replication and chaining, DSAs collectively provide a widely distributed service that appears as one Directory to the end user population in all domains



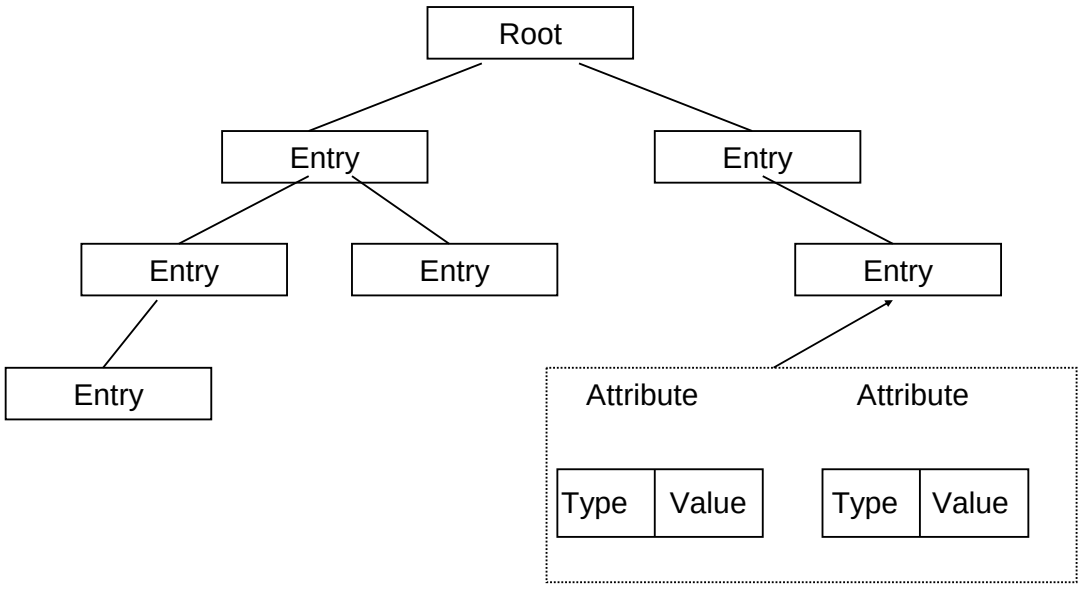
Information Model

- *determines the form and character of information in the directory;*
- *the information model is centered around **entries**, which are composed of attributes; each entry is associated to an object class which describes different attributes the entry is expected to contain*
- ***object class:** defines which attributes are required and allowed in an entry;*
- *each **attribute** has a type and one or more values;*
- *entries are arranged in a tree structure and divided among servers in a geographical and organizational distribution; each entry has a DN*

Common Name	**Rebecca Montanari**
Telephone Number	**02/57401235**

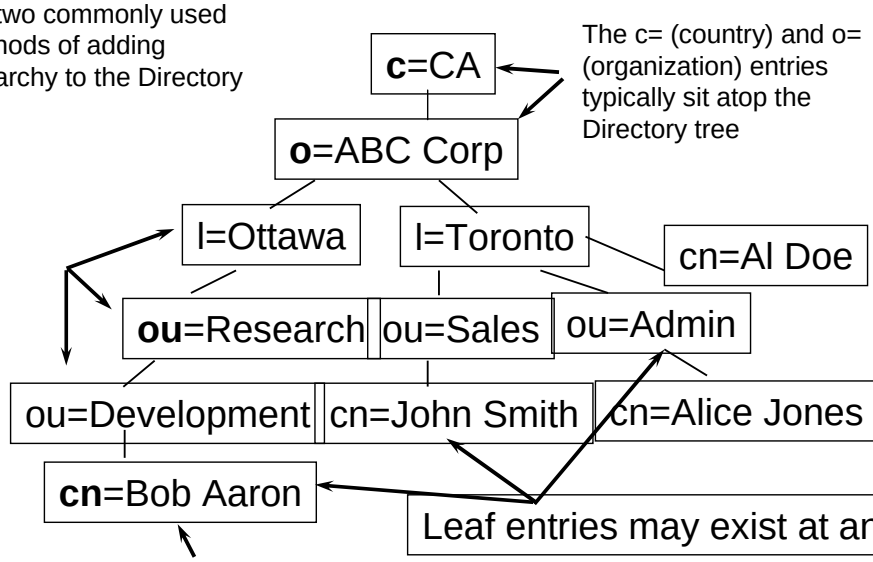
Entry

Directory Structure (1.)



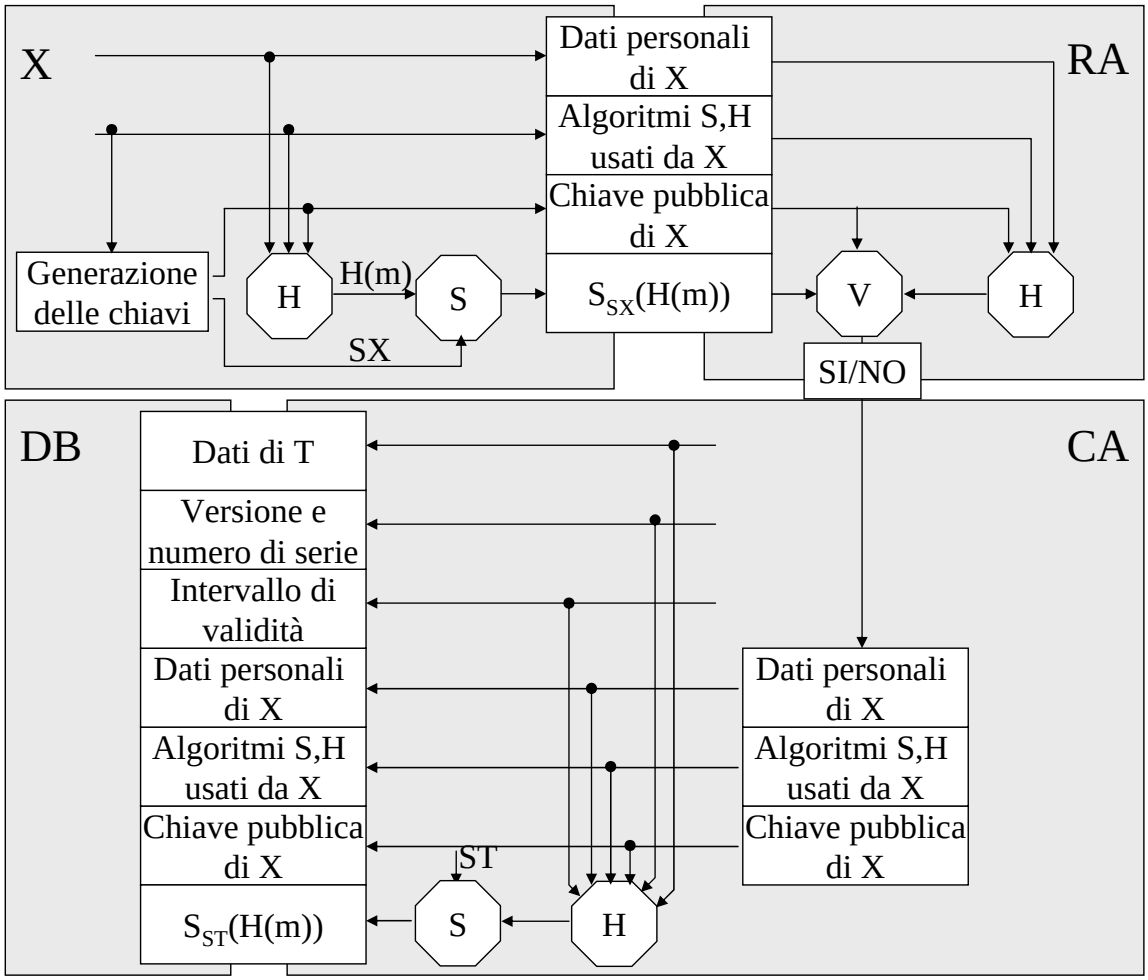
Directory Structure (2.)

l= (locality) and ou= (organizational unit) entries are two commonly used methods of adding hierarchy to the Directory



The c= (country) and o= (organization) entries typically sit atop the Directory tree

The distinguished name (DN) is still just a sequence of RDNs:
cn=Bob Aaron,ou=Development,ou=Research,l=Ottawa,o=ABC Corp,c=CA



Protocolli di Gestione

Initial registration/certification:

- Centralised model schema
- Three-party model schema

Centralised schema:

- end-entity, CA;
- initiation occurs at the certifying CA;
- "key generation" occurs at the certifying CA;

The only message required is sent from the CA to the end entity containing the entire personal security environment for the end-entity

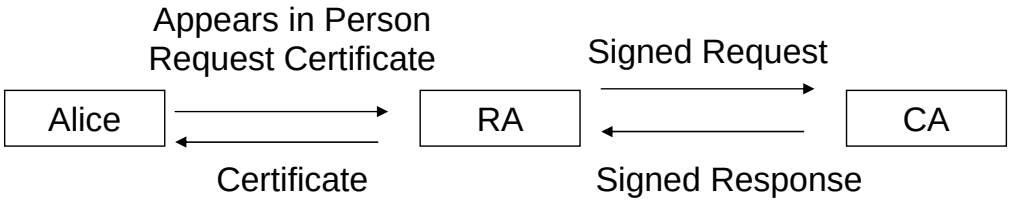
Three-party model schema:

- end-entity, RA, CA;
- different solutions

Protocolli di Gestione

Three-party model schema solutions:

1. Alice generates keys and presents physical/electronic credentials to RA; RA reviews the credentials and forwards the information to the CA, the CA generates the certificate and reurns it to the RA

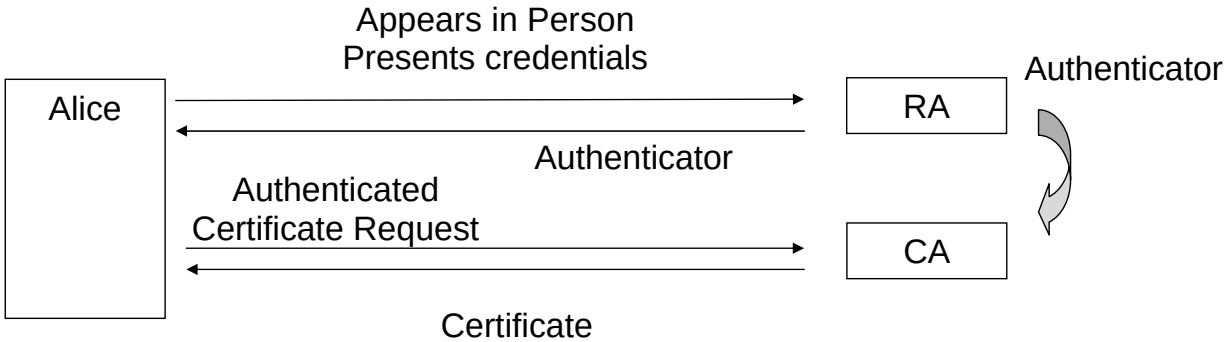


1a. Alice presents physical credentials to RA which the RA reviews; the RA provides Alice with a cryptographic module. Alice commands the module to generate the keys; the module gives the RA the public key and the RA on behalf of Alice requests a certificate

1b. RA obtains a cryptographic module, such as a smartcard, for Alice. The RA generates the keys on the module and then RA requests a certificate from the CA. The CA returns to RA the certificate. Alice comes to the RA, presents credentials and receives the module. Alice changes the password so that no one else can access the private key

Protocolli di Gestione

2. Alice presents physical/electronic credentials to RA and obtains an authenticator; Alice returns to her system and generates keys; Alice requests a certificate using the authenicator



Authenticated Request

End entity RA/CA

out-of-band distribution of Initial Authentication Key (IAK) and reference value (RA/CA -> EE)

Key generation

Creation of certification request

Protect request with IAK

(registration token control value in the certificate request message)

-->>--certification request-->>

<<--certification response--<<

-->>--confirmation message-->>

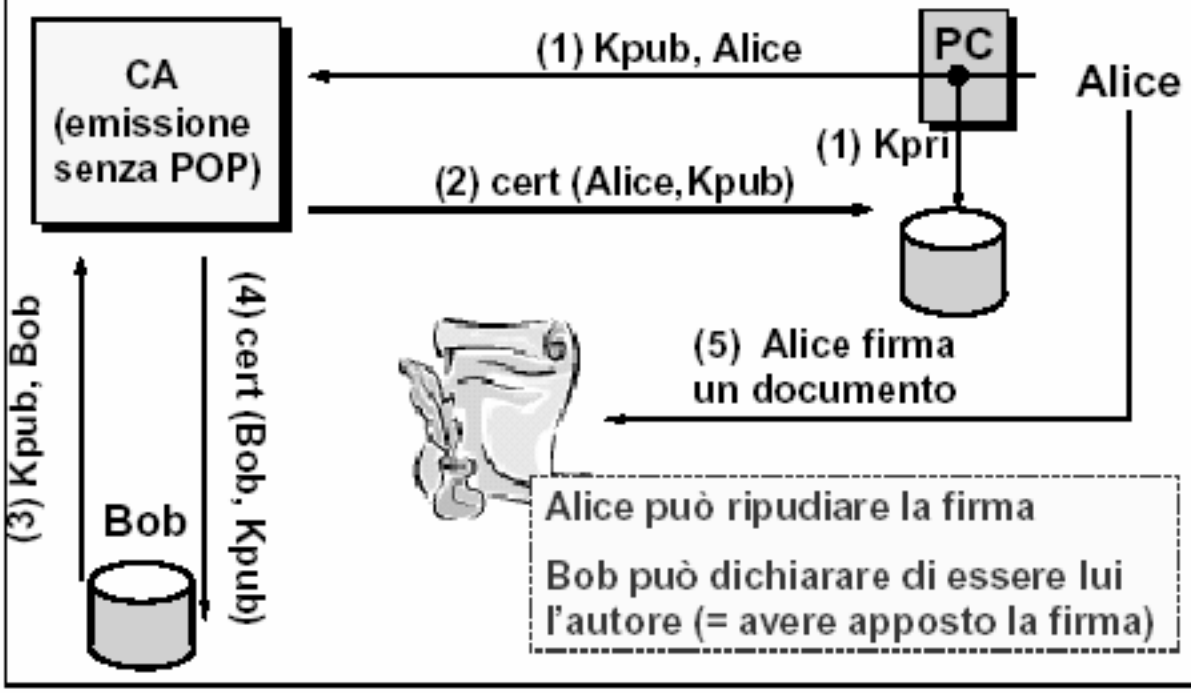
Protocolli di Gestione

3. Alice generates keys and requests a certificate. The CA queues the certificate requests for RA review. The RA reviews the client information and grants/denies the request. If approved the CA generates the certificate.

Proof-Of-Possession (POP)

- **POP = la CA ha sufficienti garanzie circa il possesso della chiave privata da parte dell'entità che richiede un certificato (il Subject)**
- **emissione di certificati senza POP può permettere vari attacchi**
  - situazione diversa a seconda dell'uso della chiave (POP fondamentale per garantire il non-ripudio)
  - POP non sempre critico nel caso di cifratura

Assenza di POP-possibili rischi



Contromisure

- **metodo migliore: POP a tempo di firma**
  - il firmatario inserisce un riferimento al certificato (es. un hash) tra le informazioni firmate
  - la firma è funzione quindi anche del certificato
  - attualmente non supportato dai protocolli di sicurezza
- **soluzione alternativa: la CA emette un certificato solo se ha la prova che il richiedente possiede la chiave privata**

Metodi per il POP (I)

- **metodi OOB**
  - chiavi generate da CA/RA e consegnate in token sicuri (es. smart-card, USB crypto-token); fa quindi fede il possesso del token
  - politiche di key-recovery/key-backup (molto rischioso!!!): la CA mantiene una copia di tutte le chiave private – come le protegge efficacemente?

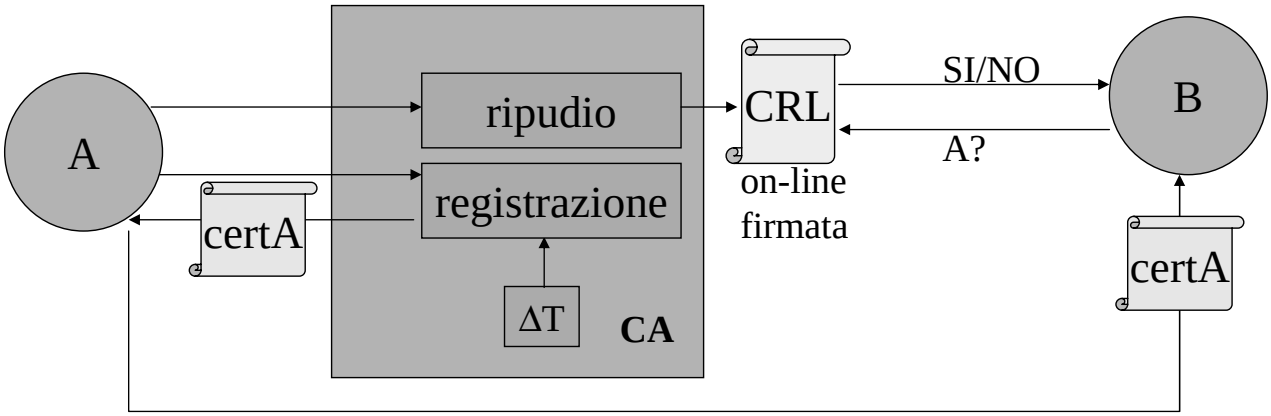
Metodi per il POP (II)

- **metodi on-line**
  - per chiavi di firma e cifratura: possibile usare formati auto-firmati (PKCS-10, SPKAC): la CA verifica la firma ed ottiene il POP
  - per chiavi di cifratura (no firma)
    - utilizzo di protocolli challenge - response che comportino un'operazione di decifratura (=uso della chiave privata)
    - certificato restitituito in forma cifrata (successiva revoca se il certificato non è usato)

Vita e ripudio di una chiave pubblica

R29: "*Quando uno ha il sospetto, o la certezza, che la sua chiave segreta sia stata violata, deve*
• *rinunciare ad impiegarla,*
• *notificare immediatamente il ripudio alla CA*
• *registrarne una nuova*".
Per prevenire il rischio CA da una vita limitata ad ogni chiave.

Ripudio anche per cambio di ruolo e di Società



Modelli di Notifica di Revoca

- **pull method**
- **push model**
- **offline status checking**
- **online status checking**

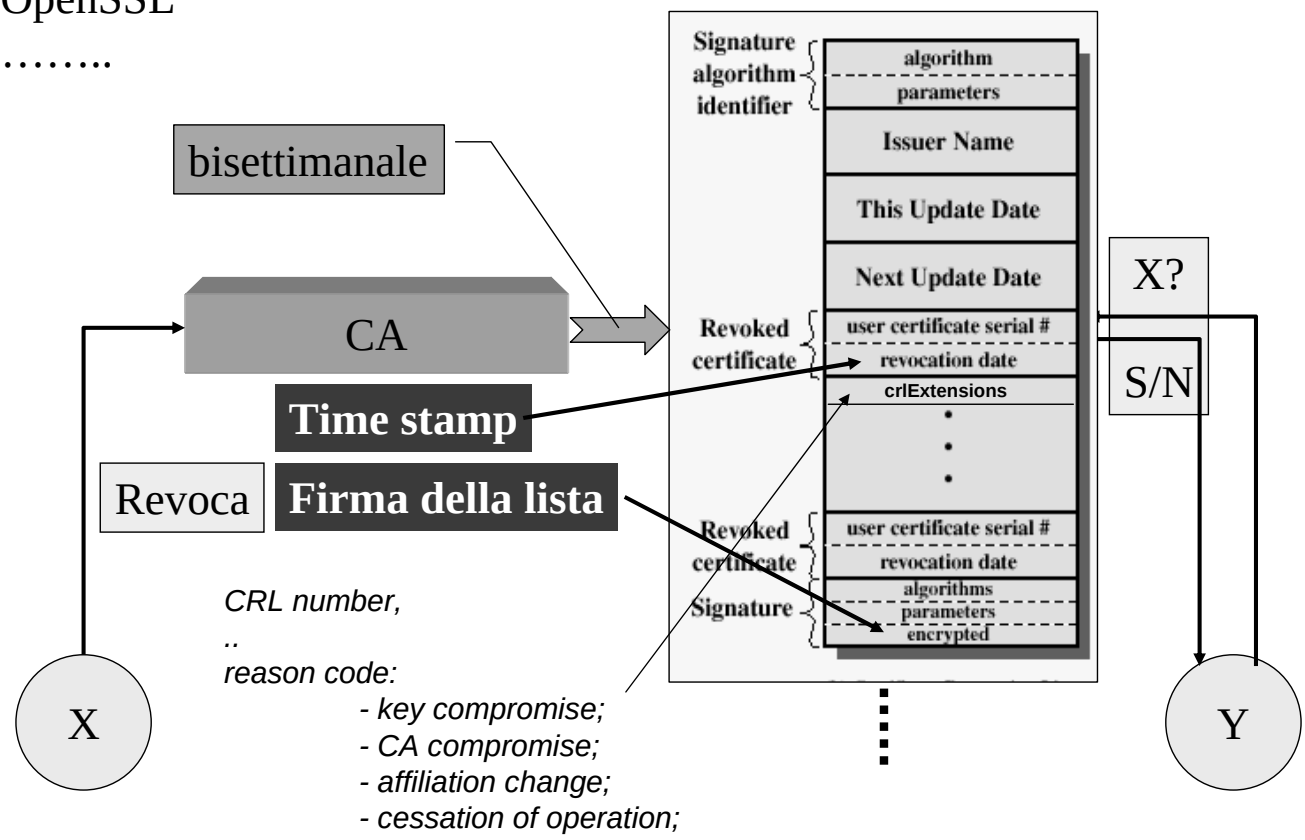
Schemi di Notifica della Revoca:

Schemi off-line:
Certificate Revocation List
Certificate Revocation Tree
…

Schemi on-line:
On-line Certificate Status Protocol

RFC 2510
OpenCA
OpenSSL
……..

CRL



Dimensione CRL

Hp: frequenza di revoca circa del 10 per cento all'anno,
certificati con validità temporale di due anni,
popolazione stabile di circa 100000 certificati

⇩

dimensione media di una CRL circa 20000 entries

Gestione efficiente delle CRL

- **problema: le CRL possono diventare molto grosse e quindi onerose da scaricare e da esaminare**
- **varie soluzioni:**
  - eliminare la revoca dopo la prima CRL successiva alla scadenza del certificato
  - pubblicare CRL complete (Base CRL) e poi solo le differenze (Delta CRL)
  - partizionare le CRL in tanti gruppi (es. per ogni mille certificati emessi) usando CRL DP

Come ridurre la dimensione delle CRL?

- **partizionamento delle CRL (usando opportunamente il CRLDP):**
  - cert con SN < 1000 hanno CRLDP= crl_1_1000.der
  - cert con 1000 < SN < 2000 hanno CRLDP= crl_1001_2000.der
- **usando deltaCRL**
  - una CRL base (n. N)
  - una o più delta CRL (differenza rispetto a N)

Estensioni delle CRL

- ***general extensions:***

  - delta CRL

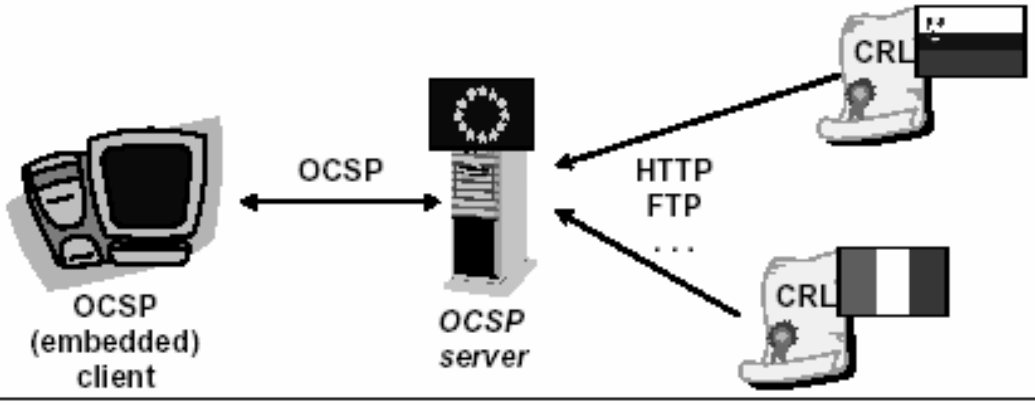
  - Issuing Distribution Point (certificates should include the CRL distribution point extension to help certificate users locate the CRL)

OCSP

- **RFC-2560: On-line Certificate Status Protocol**
- **standard IETF-PKIX per verificare in linea se un certificato è valido:**
  - good
  - revoked
    - revocationTime
    - revocationReason
  - unknown
- **risposte firmate dal server (non dalla CA!)**
- **certificato del server non verificabile con OCSP!**

Architettura di OCSP

- **possibili risposte pre-calcolate**
  - diminuisce il carico sul server … ma rende possibili attacchi di tipo replay!
- **possibile attingere informazioni non da CRL**



Modelli di OCSP responder

- **Trusted Responder**
  - il server OCSP firma le risposte con una coppia chiave:cert indipendente dalla CA per cui sta rispondendo
  - responder aziendale o TTP pagata dagli utenti
- **Delegated Responder**
  - il server OCSP firma le risposte con una coppia chiave:cert diversa in base alla CA per cui sta rispondendo
  - TTP pagata dalle CA

Performance Evaluation Criteria

- Timeliness
- Involved computational load
- Communication traffic induced on the network

Parametro	Parole chiave	
Prestazioni	Lato Amministratore	Picco di Carico e Picco di Richiesta Carico Medio e Richiesta Media Distribuzione del Carico Ritardo Dimensioni
	Lato Utente	Dimensioni Ritardo Massimo Carico Computazionale Banda
Tempestività		Tempo Massimo tra revoca e distribuzione
Scalabilità	Lato Amministratore	Complessità dello schema
Sicurezza		Autenticità Integrità Confidenzialità Non-Ripudio
Standard		Standard Proprietario Teorico Implementato
Espressività		Granularità dell'informazione di revoca
Gestione dello schema	Lato Amministratore	Automatizzato Archiviazione sicura Complessità
On-line vs. Off-line	Lato Amministratore	Frequenza delle connessioni

Directory and PKI

Attributes:

- userCertificate
- cACertificate
- certificateRevocationlist
- authorityRevocationList
- deltaRevocationlist
- crossCertificatePairs

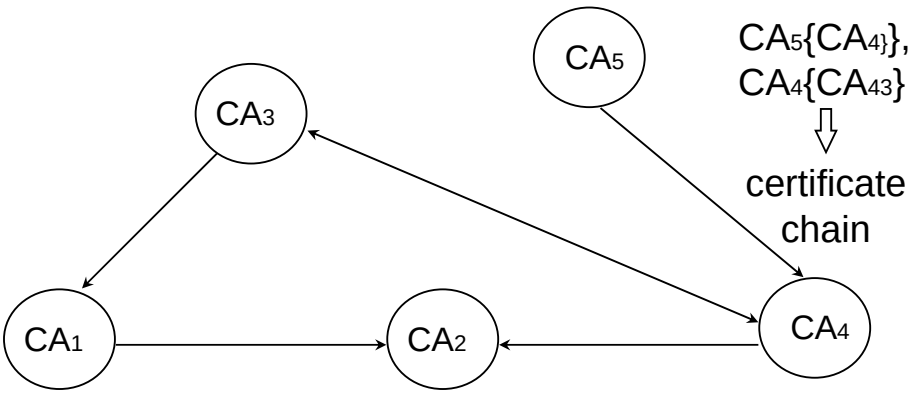
Three object classes:

- pkiUser
- pkiCA
- cRLDistributionPoint

Trust Models

certificate chains and certification paths:

$A\{P_5\} \Rightarrow B\{P_3\}$



- ***centralized*** trust model;
- ***distributed*** trust model

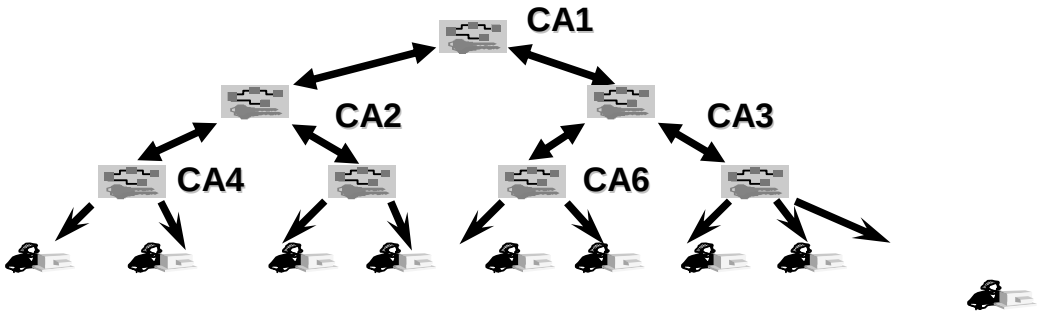
Certification Path Discovery

- the certification path discovery problem is as follows: I need to find a certification path between a particular remote user's public key and any one of the set of root public key that I already know securely
- given a CA's name, a service to retrieve certificates for that CA's public keys issued by other CAs, it is possible to find a certification path by working back progressively from the target user's certificate toward a root key, as follows:
  - **step 1**: given a certificate issued by CA X, determine the set of CAs that have issued certificates for the public key of X;
  - **step 2:** if one of the CAs from the Step 1 is a known root authority, the required certification path is found, otherwise proceed to Step3;
  - **step 3:** for each CA found in Step 1, repeat the Step 1 procedure, treating that CA as CA X

Certification Path Validation

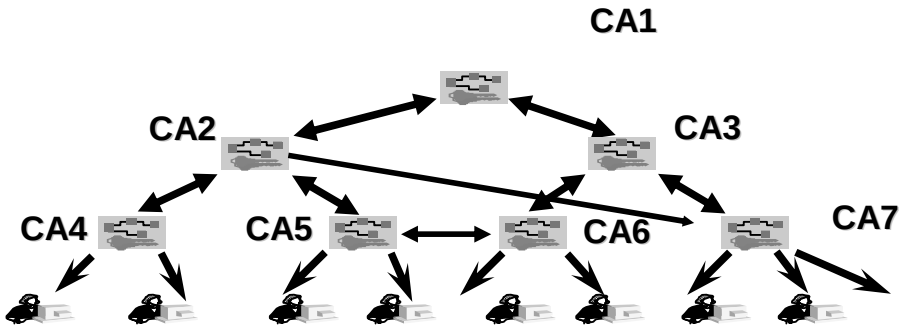
- given that a suitable certification path has been found, it is then necessary to validate that path. This involves such actions as:
  - verifying the digital signature on each certificate;
  - checking that the names in the certificates are consistent with a valid certification path, that is, **the subject of every certificate is the issuer of the next certificate**;
  - checking that the validity periods of all certificates correctly span the time for which validity is being checked;
  - checking that each certificate has not been revoked. This may be a complex process;
  - checking that the required certificate policies are indicated in the certificates;

General Hierarchical Structure



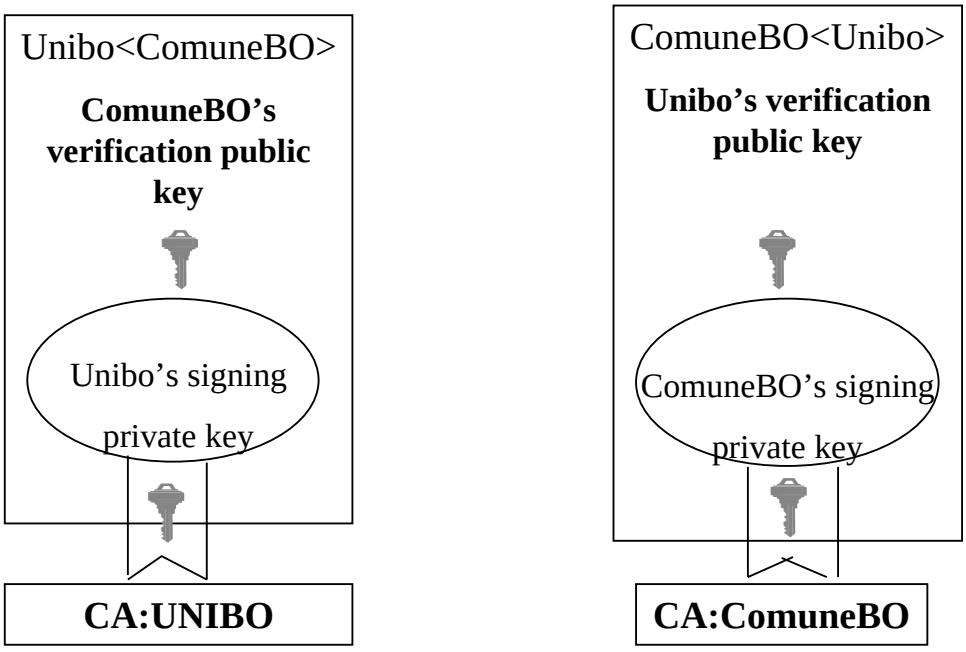
- it is easy to construct a certification path between any pair of end-entities, regardless of how each end-entity determines which CA it is prepared to accept as root CA;
- this model scales reasonably well; provides means for constructing reasonably short certification paths;
- complicating factor is trust

General Hierarchical Structure with additional links

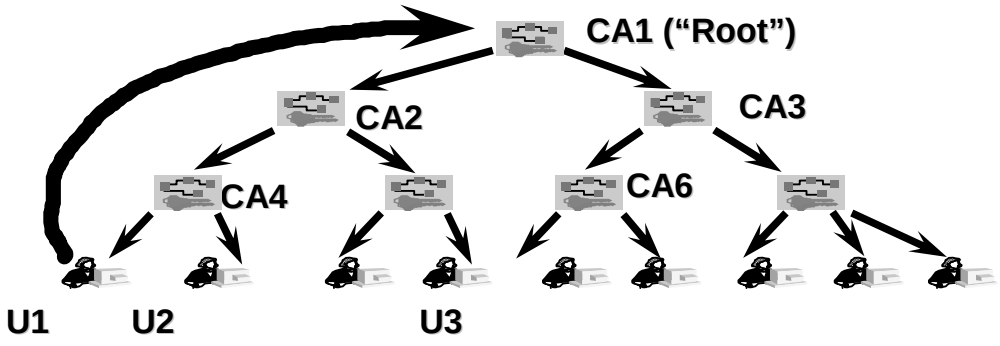


- added links are called cross-certificates

Cross-Certificates



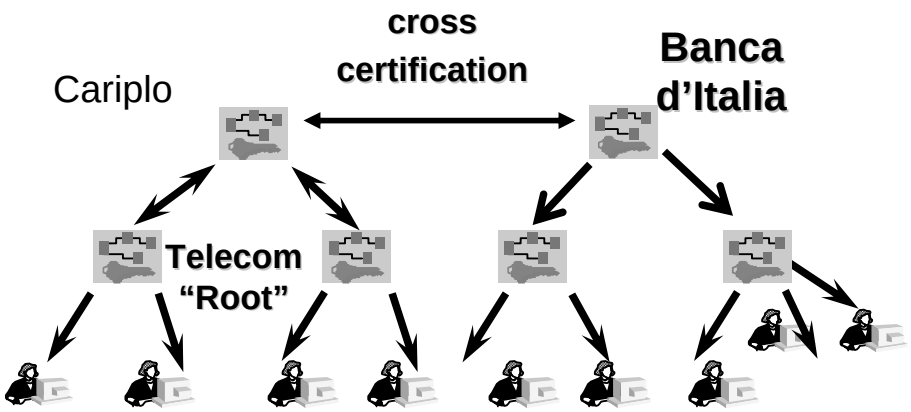
Top-down Hierarchical Structure



Drawbacks:

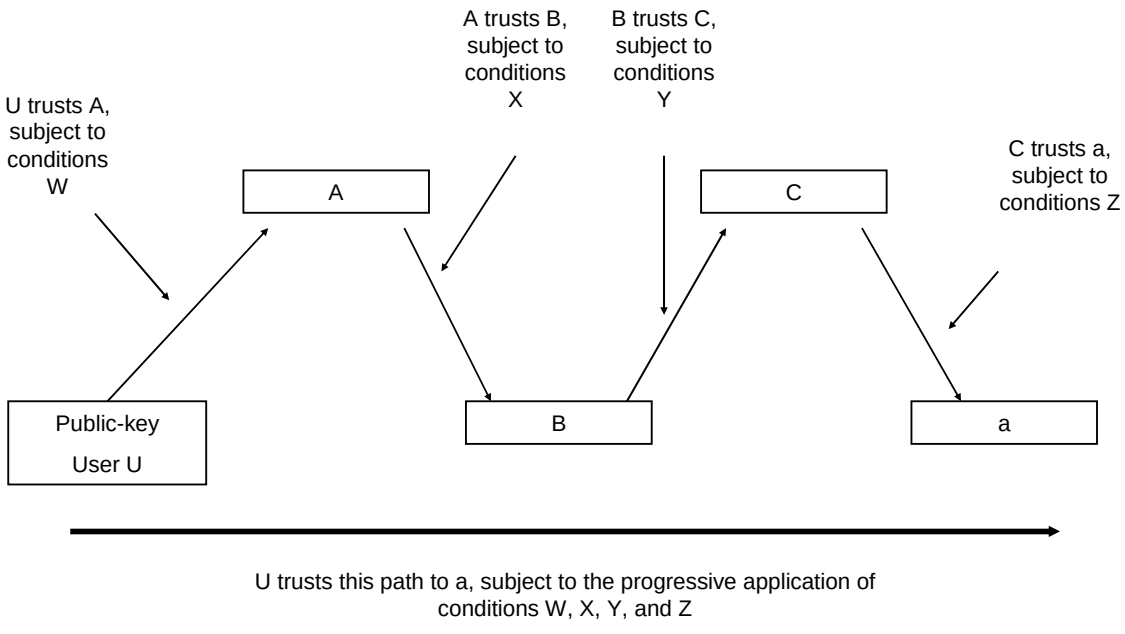
- all trust depends on the root key;
- certificate chains also for two entities on the same CA;
- certificate chains long in deep hierarchies.

Distributed Trust Model



Trust is rooted at the CA close to the end users

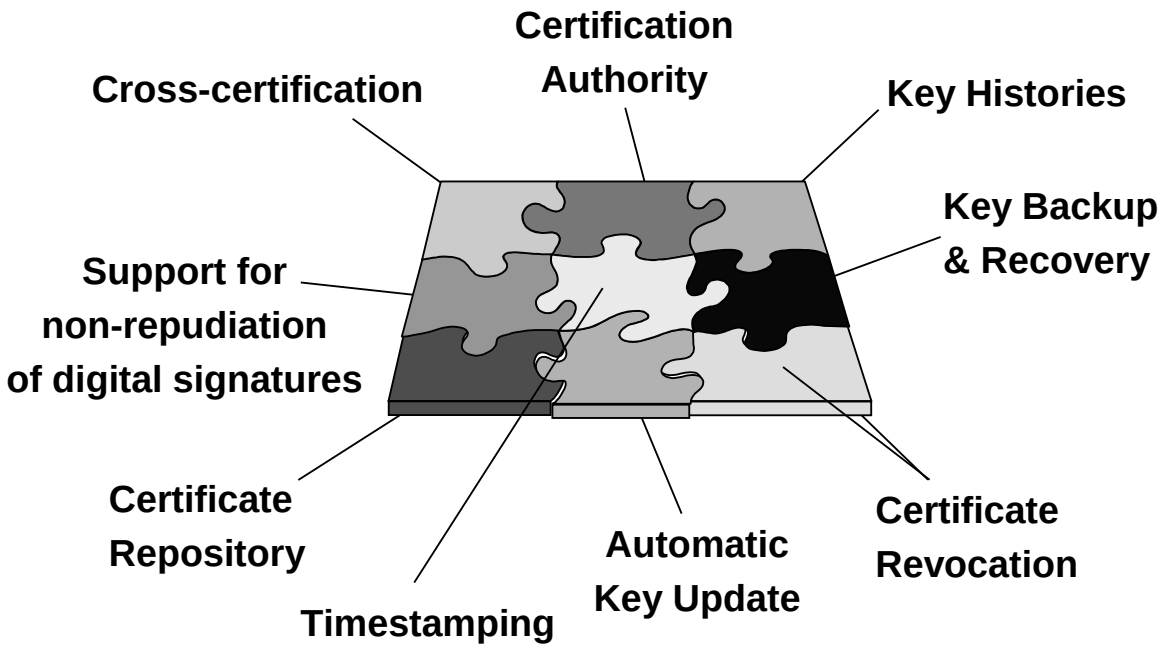
Progressive-Constraint Trust Model



X.509 Certificate Policies

- **certificate policy:** a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements;
- **policy mapping:** only applies to cross-certificates; provides a mechanism for the signing CA to map its policies to the policies of the CA specified in the cross-certificate
- **policy constraints:** it is used in cross-certificates; the administrator can specify the set of acceptable policies in a certificate chain extending from a cross-certificate; can specify whether or not all certificates in a chain must meet a specific policy;
- ……

Quando un sistema è una PKI?



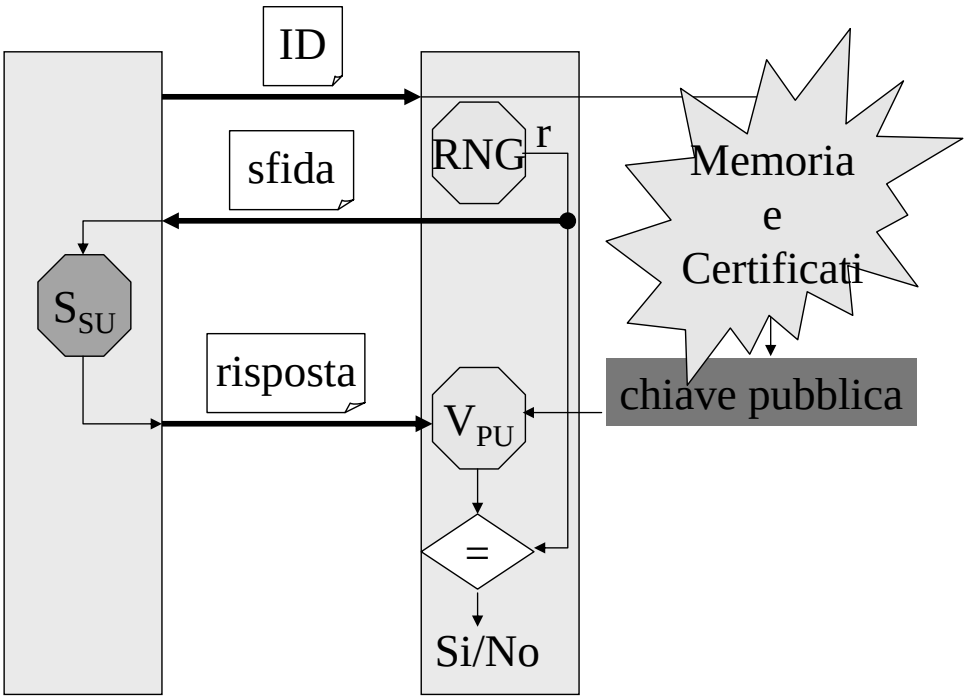
Problemi di PKI

- RA sempre disponibile
- CA rapida anche nella gestione della CRL
- Collo di bottiglia (n° max di utenti)
- Ente degno di fiducia
- Interrogazione della CRL
- Vita della chiave di firma

Identificazione con C/R

Identificando Verificatore

Algoritmo di firma
•con recupero
•con appendice



Identificazione unilaterale

1 - A ← B: RB
2 - A → B: c = S_{SA}(RB)
3 - V_{PA}(c) = RB ?

Attacco con testo scelto

2 - A → B: RA||S_{SA} (RA||RB)

Chiavi diverse

Attacco dell'uomo in mezzo

2 - A → B: CERT(PA,T)||RA||S_{SA}(RA||RB)

Il problema dei GM degli scacchi

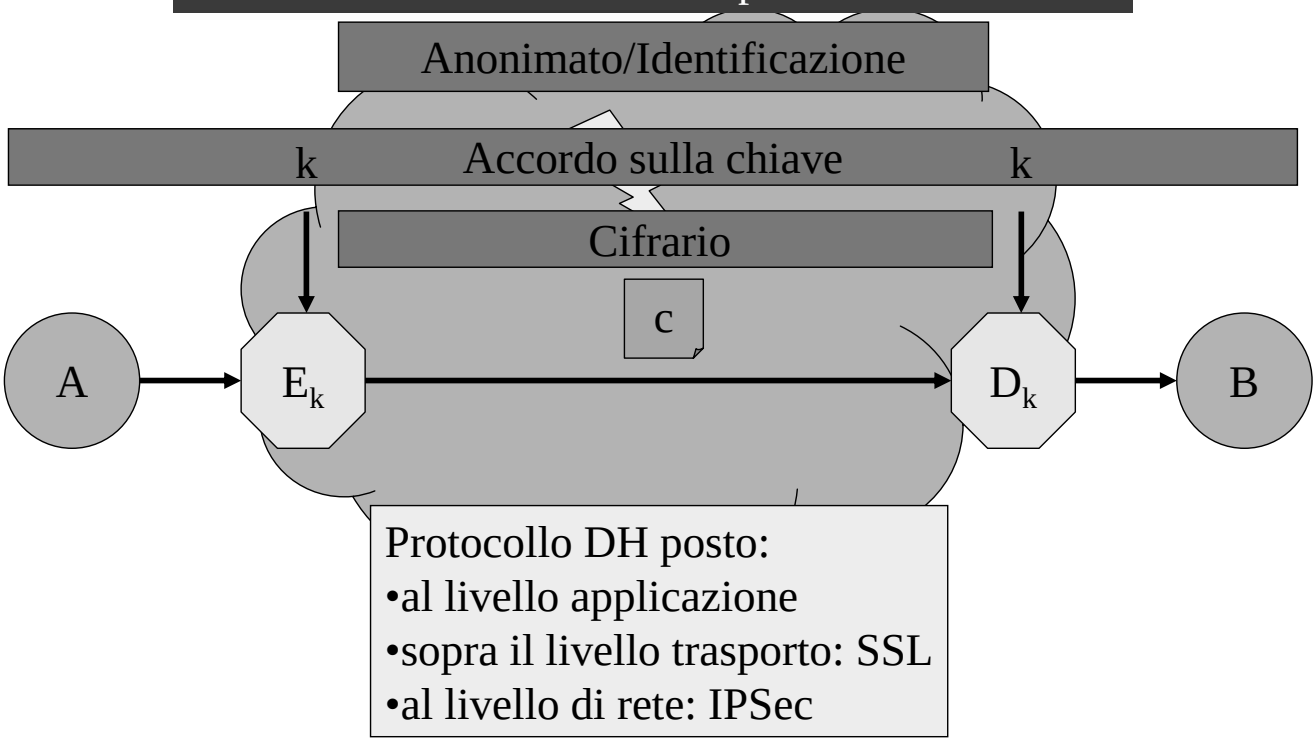
2 - A → B: CERT(PA,T)||RA||B||S_{SA}(RA||RB||B)
τ τ

Identificazione reciproca

1 - A ← B: τB1
2 - A → B: CERT(PA,T)|| τA||B||S_{SA}(τA||τB1||B)
3 - A ← B: CERT(PB,T')|| τB2||S_{SB}(τB2|| τA||A)

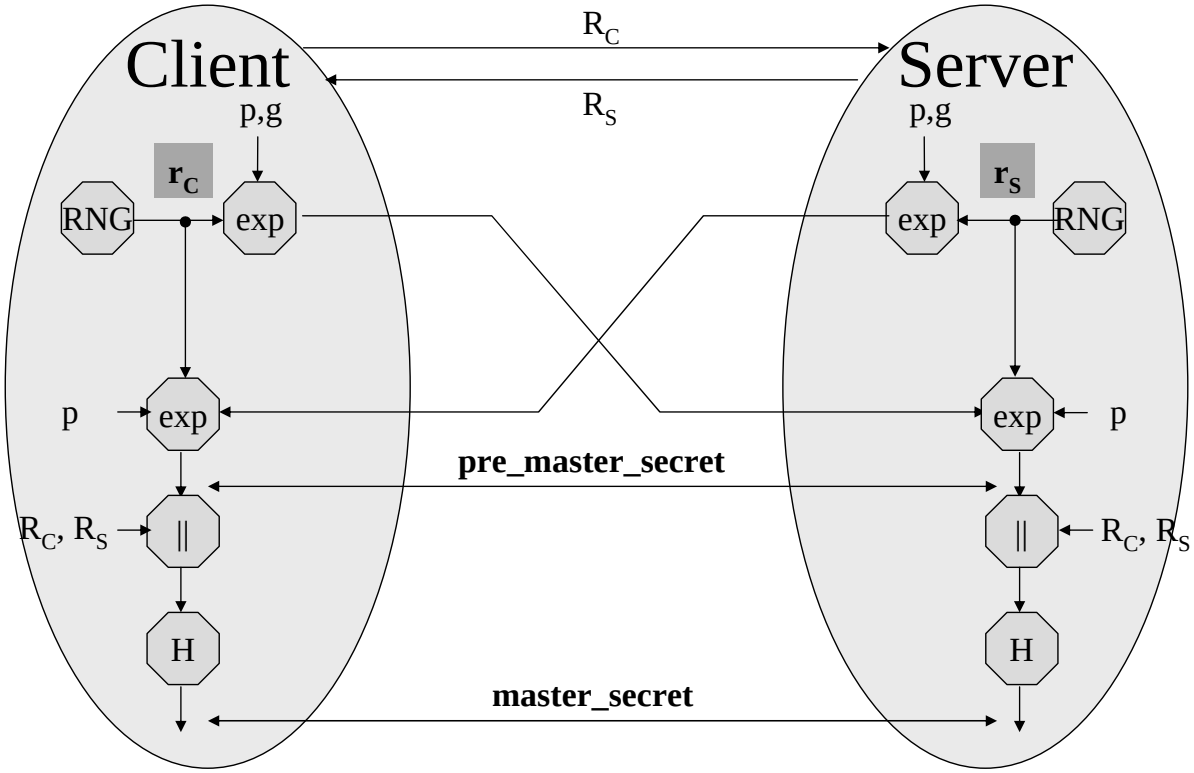
I certificati e le reti sicure

Problema: A e B vogliono scambiarsi informazioni riservate in assenza di accordi precedenti

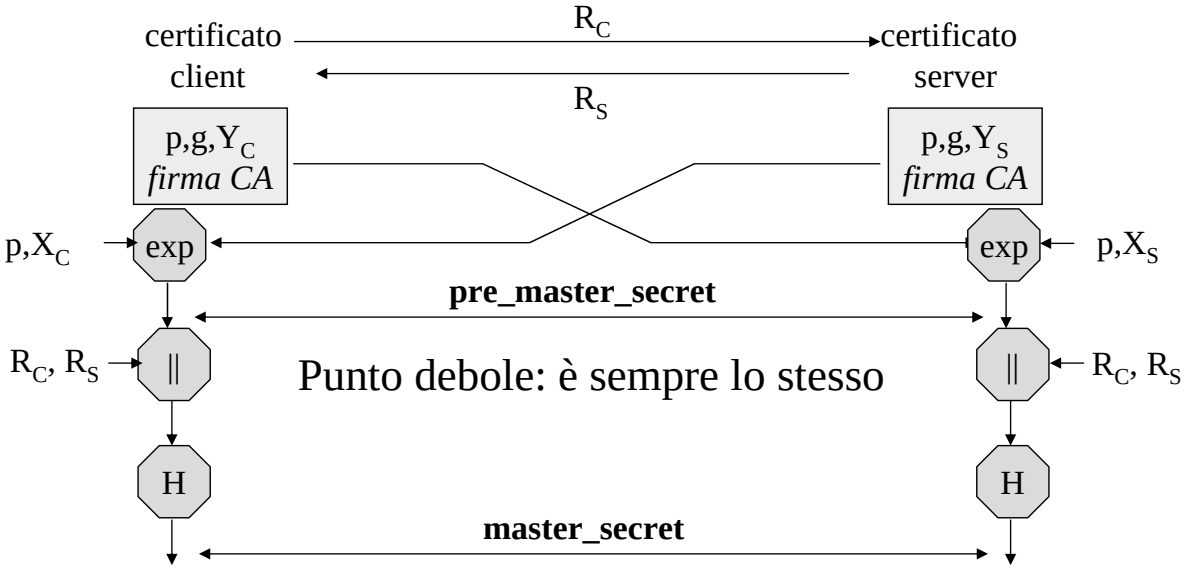


Protocollo DH posto:
- al livello applicazione
- sopra il livello trasporto: SSL
- al livello di rete: IPSec

L'accordo sul segreto: anonimous Diffie-Hellman



L’accordo sul segreto: fixed Diffie-Hellman



L’accordo sul segreto: ephemeral Diffie-Hellman