Protocolli di Rete

Sabrina De Capitani di Vimercati

decapita@ing.unibs.it.

DEA - Università di Brescia

Ultimi Mattoni: La Firma Digitale

A cosa serve? Il destinatario di un crittogramma può convincere una terza persona della sua provenienza e che il messaggio non ha subito variazioni

Quando? Non fiducia reciproca tra Alice e Bob: Alice non deve essere in grado di negare l'invio di un messaggio e Bob non deve essere in grado di modificare il messaggio e di attribuirlo a Alice

Desiderata della Firma Digitale

- La firma digitale deve essere facilmente prodotta dal legittimo firmatario
- La firma è autentica e quindi chi la riceve è convinto della sua originalità
- La firma non deve poter essere falsificata
- La firma non deve poter essere riutilizzata
- La firma non deve poter essere ripudiata
- Il documento firmato non deve poter essere alterato
- Chiunque può facilmente verificare una firma

Firma Digitale con Cifrari Simmetrici o Asimmetrici?

Cifrari Simmetrici Non sono appropriati per la realizzazione della firma digitale perché gli algoritmi proposti sono costosi sia dal punto di vista della computazione che dal punto di vista della trasmissione

Cifrari Asimmetrici Si prestano molto bene per la definizione di firme digitali perché sono semplice ed economici

RSA per Firma Digitale

Sia (e, n) la chiave pubblica di Alice e (d, n) la corrispondente chiave privata

- la firma di un messaggio m è data da $f = m^d \mod n$
- la verifica della firma f consiste nel verificare se $m = f^e \mod n$

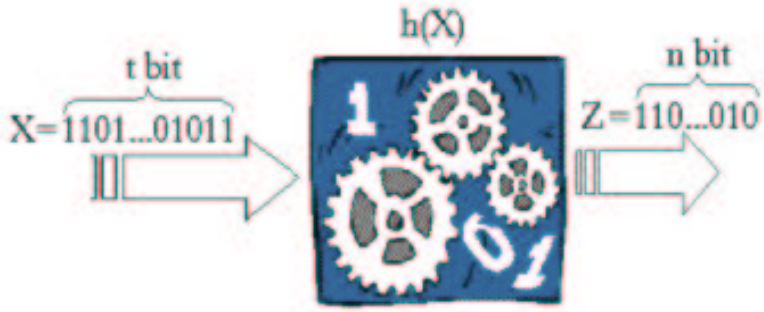
Come calcolare la firma di m se $m > n$?

Firma Digitale di Grandi Messaggi

- Prima soluzione: si decompone m in più blocchi $m_i < n$, si firma ciascun blocco singolarmente e si concatenano le firme di ciascun blocco
  $\Rightarrow$ poco efficiente e la permutazione delle firme genera una nuova firma
- Seconda soluzione: si calcola $h(m)$ e si firma il valore hash così ottenuto
  $\Rightarrow$ efficiente

Funzioni Hash

- Le funzioni hash hanno assunto un ruolo fondamentale per la garanzia dell'integrità dei dati e l'autenticazione dei messaggi
- Una funzione hash prende in input un messaggio e restituisce un output denominato codice hash o semplicemente hash



Funzioni Hash One-Way

- Il concetto di funzione hash viene combinato con la proprietà one-way

  $\Rightarrow$ importante per la realizzazione delle firme digitali

- Le proprietà che caratterizzano una funzione one-way $f : X \rightarrow Y$ sono:
  - dato un elemento $x \in X$ è computazionalmente facile calcolare $f(x)$ (tempo polinomiale nella dimensione dell'input)
  - dato un elemento $y \in Y$ è computazionalmente difficile trovare un x tale che $y = f(x)$ (tempo esponenziale nella dimensione dell'input)

Caratteristiche Funzioni Hash

One-Way

1. Deve essere computazionalmente difficile trovare una coppia x e x' in X tale per cui $f(x) = f(x')$ (claw-free)
2. $|Y| << |X|$

Le funzioni hash one-way non possono essere usate per la definizione di sistemi asimmetrici a causa della condizione 2. (la decifrazione non è univoca)

Secure Shell (SSH)

SSH consente di stabilire una connessione crittata tra due sistemi ed include:

- connessioni remote;
- trasferimento di file sicuro;
- . . . . . .

La connessione che si instaura è del tipo client-server, quindi è necessario vedere il funzionamento di un SSH client e SSH server

SSH Client (1)

Due modalità di funzionamento:

1. SSH server che ammette autenticazione via password:
   $\rightarrow$ digitare ssh hostname oppure ssh username@hostname
2. SSH server che non ammette autenticazione via password:
   $\rightarrow$ è necessario creare una coppia di chiavi

SSH Client: Generazione di Chiavi

Sotto sistemi UNIX-like si usa il programma ssh-keygen -d

- l'utente inserisce una passphrase
- il programma genera una coppia di chiavi pubblica e privata (identity, identity.pub)
- la chiave pubblica identity.pub deve essere memorizzata sul server nel file .ssh/authorized_keys2
- la chiave pubblica deve anche essere autorizzata sulla macchina originale (~/.ssh/authorized_keys2)

Esempio di Chiave Pubblica

—- BEGIN SSH2 PUBLIC KEY —-
Subject:
Comment: my public key

AAAAB3NzaC1kc3MAAACBAKoxPsYlv8Nu+fncH2ouLiqkuUNGIJo8iZaHdpDABAvCvLZn
jFPUN+SGPtzP9XtW++2q8khlapMUVJS0OyFWgl0ROZwZDApr2olQK+vNsUC6ZwuUDRPV
fYaqFCHrjzNBHqgmZV9qBtngYD19fGcpaq1xvHgKJFtPeQOPaG3Gt64FAAAAFQCJfkGZ
e3alvQDU8L1AVebTUFi8OwAAAIBk9ZqNG1XQizw4ValQXREczlIN946Te/1pKUZpau3W
iiDAxTFlK8FdE2714pSV3NVkWC4xlQ3x7wa6AUXIhPdLKtiUhTxtctm1epPQS+RZKrRI
XjwKL71EO7UY+b8EOAC2jBNIRtYRy0Kxsp/NQ0YYzJPfn7bqhZvWC7uiC+D+ZwAAAIEA
mx0ZYo5jENA0IinXGpc6pYH18ywZ8CCI2QtPeSGP4OxxOusNdPskqBTe5wHjsZSiQr1g
b7TCmH8Tr50Zx+EJ/XGBU4XoWBJDifP/6Bwryejo3wwjh9d4gchaoZNvIXuHTCYLNPFo
RKPx3cBXHJZ27khllsjzta53BxLppfk6TtQ=

—- END SSH2 PUBLIC KEY —-

SSH Server: Il File di Configurazione

- Il demone SSH si chiama sshd
- Il file di configurazione (sshd_config) si trova in /etc/ssh ed è composto da coppie del tipo: “ValueName Value”
  Es.
  PasswordAuthentication yes
  RSAAuthentication yes
  HostKey /etc/ssh/ssh_host_key
  KeyRegenerationInterval 3600

SSH: La Connessione (1)

La connessione ha inizio digitando ssh hostname oppure ssh username@hostname

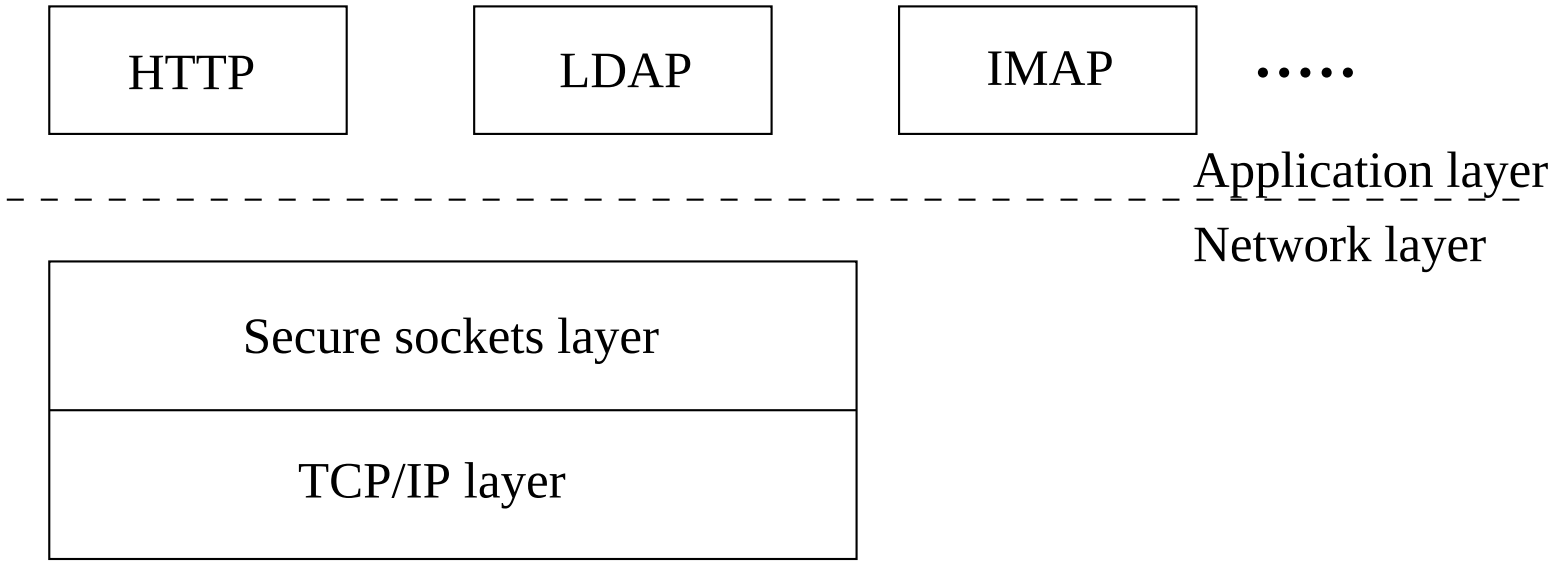
- Quando il client si connette al server, il server manda la propria chiave pubblica
- La prima volta che il client si connette al server, il client non ha modo di sapere se è la chiave pubblica del server
  $\rightarrow$ visualizzato un messaggio di warning
- Il server deve inoltre autenticare il client e genera un “challenge” crittato con la chiave pubblica del client e poi lo invia

SSH: La Connessione (2)

- L'utente deve digitare la passphrase (non viene trasmessa, serve solo localmente a decrittare la chiave privata)
- Il client usa la chiave privata per decrittare il "challenge" e per rispedirla al server crittata con la sua chiave privata
- Il client genera una chiave di sessione la critta con la chiave pubblica del server e la invia
- Il client riceve un msg di conferma da parte del server
- Il client e server si scambiano dati usando la chiave di sessione

Protocollo SSL

Opera tra il livello della rete ed il livello delle applicazioni



Componenti di SSL

- SSL Handshake protocol: permette di stabilire un canale sicuro tra le parti
  - autenticazione del server al client
  - scelta dei protocolli crittografici da usare durante la sessione
  - autenticazione opzionale del client al server
  - crittografia asimmetrica per generare un “segreto”
  - connessione SSL crittata
- SSL Application Data Protocol: usato per lo scambio di dati lungo il canale

Ciphersuite di SSL

Client e server possono supportare diversi cifrari; dipende dalla versione di SSL adottata, restrizioni governative sull'esportazione di SSL

- Algoritmo per lo scambio di chiavi
- Algoritmo per l'autenticazione
- Algoritmo per la cifratura simmetrica
- Algoritmo per il MAC

Algoritmi usati da SSL

- **DES**, cifrario simmetrico usato negli USA
- **DSA**, Digital Signature Algorithm, fa parte del digital authentication standard usato negli USA
- **KEA**, Key Exchange Algorithm, un algoritmo usato per lo scambio di chiavi negli USA
- **MD5**, Message Digest algorithm
- **RC2** e **RC4**, cifrari sviluppati da Rivest per la RSA Data Security
- **RSA**, cifrario asimmetrico
- **RSA key exchange**, un algoritmo per lo scambio di chiavi per SSL e basato su RSA
- **SHA-1**, Secure Hash Algorithm, una funzione di hash usata dagli USA
- **SKIPJACK** un cifrario simmetrico implementato in FORTEZZA e usato dagli USA
- **Triple-DES**, DES applicato tre volte

Ciphersuite supportate da SSL (1)

Dalla più forte alla più debole...

Ciphersuite 1

- Permessa solo negli USA (per banche e istituzioni che manipolano dati particolarmente sensibili)
- **Triple-Des** con **SHA-1**
- Supportata sia da SSL 2.0 che da SSL 3.0

Ciphersuite supportate da SSL (2)

Ciphersuite 2

- Permessa solo negli USA
- **RC4** con 128-bit e **MD5** (SSL 2.0, 3.0)
- **RC2** con 128-bit e **MD5** (SSL 2.0, 3.0)
- **DES** e **SHA-1** (SSL 3.0)
- **DES** e **MD5** (SSL 3.0)

Ciphersuite supportate da SSL (3)

Ciphersuite 3

- Esportabile verso diversi paesi
- **RC4** con 40-bit e **MD5** (SSL 2.0, 3.0)
- **RC2** con 40-bit e **MD5** (SSL 2.0, 3.0)

Ciphersuite 4

- Fornisce autenticazione ed integrità ma non tecniche crittografiche
- **MD5** (SSL 3.0)

SSL Handshake (1)

- Il client spedisce al server le informazioni sulla versione del protocollo SSL, ciphersuite, dati random ed altre informazioni (messaggio client hello)
- Il server spedisce al client le informazioni sulla versione del protocollo SSL, ciphersuite, dati random ed altre informazioni (messaggio server hello)
- Il server spedisce il suo certificato se il client deve autenticare il server (messaggio server certificate) e se necessario richiede il certificato del client
- Il server trasmette il messaggio server hello done per indicare la fine del server hello

SSL Handshake (2)

- Il client autentica il server e se questo non è possibile l'utente viene allertato
- Il client (con la cooperazione del server e a seconda della ciphersuit scelta):
  - crea la premaster secret per la sessione
  - critta la premaster secret con la chiave pubblica del server (ottenuta dal certificato del server)
  - spedisce la premaster secret crittata al server

SSL Handshake (3)

- Opzionalmente, se il server ha richiesto l'autenticazione del client:
  - il client firma un dato unico in questo handshake e noto a client e server
  - il client spedisce il dato firmato e il suo certificato al server insieme con la premaster secret crittata
  - se il server non è in grado di autenticare il client, la sessione viene terminata
- il server decritta la premaster secret usando la propria chiave privata
- dalla premaster secret, client e server generano la master secret
- dalla master secret, client e server generano la session key
  - la session key è una chiave simmetrica usata per crittare e decrittare le informazioni scambiate durante una sessione SSL

SSL Handshake (4)

- il client spedisce al server un messaggio per informarlo che i messaggi successivi saranno crittati e spedisce un messaggio separato (crittato) che termina la parte client dell'handshake
- il server spedisce al client un messaggio per informarlo che i messaggi successivi saranno crittati e spedisce un messaggio separato (crittato) che termina la parte server dell'handshake

SSL Application Data

Ha inizio la sessione SSL

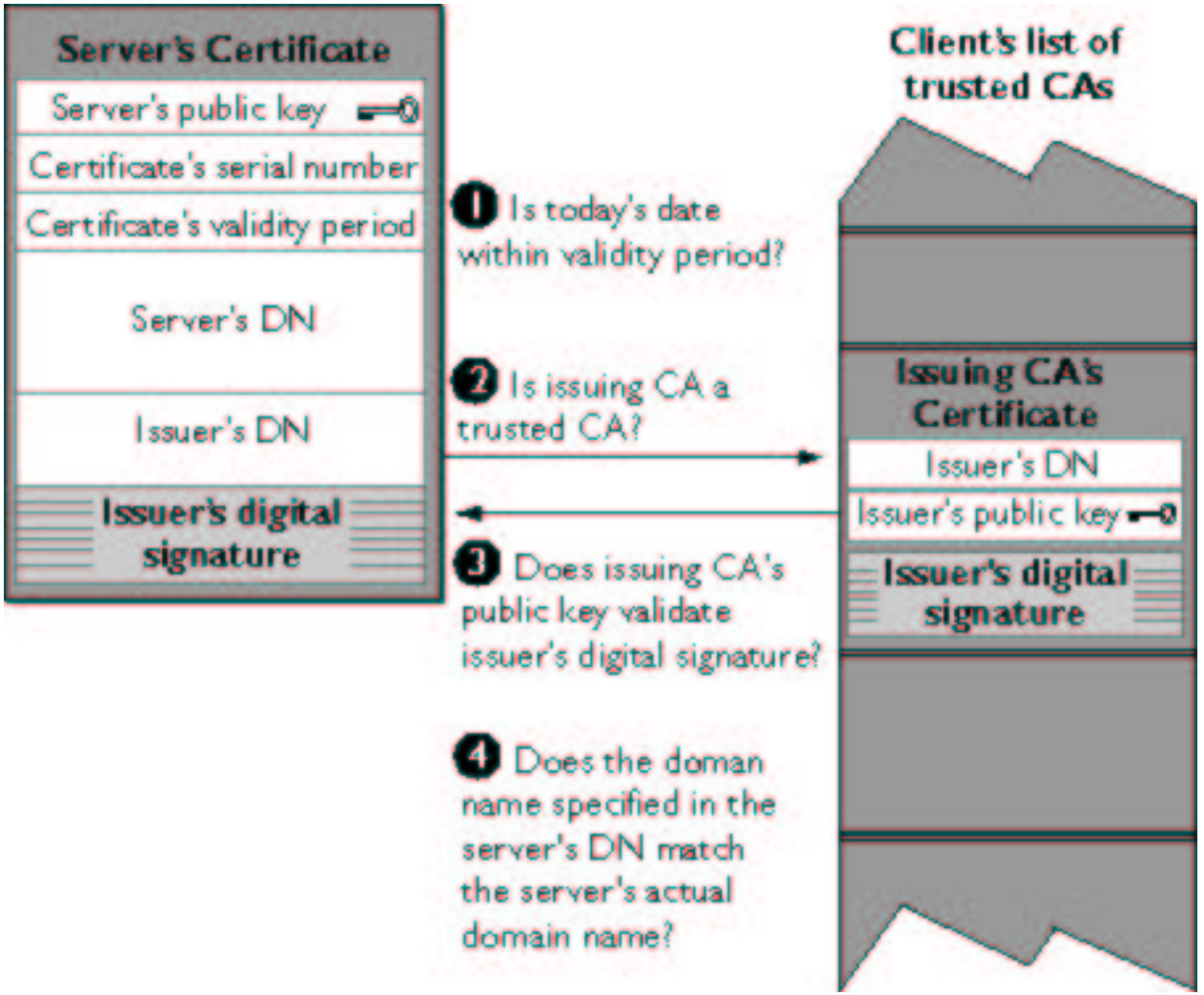
Client

- calcola il digest del messaggio da spedire
- critta il messaggio ed il digest
- spedisce il messaggio e digest crittato al server

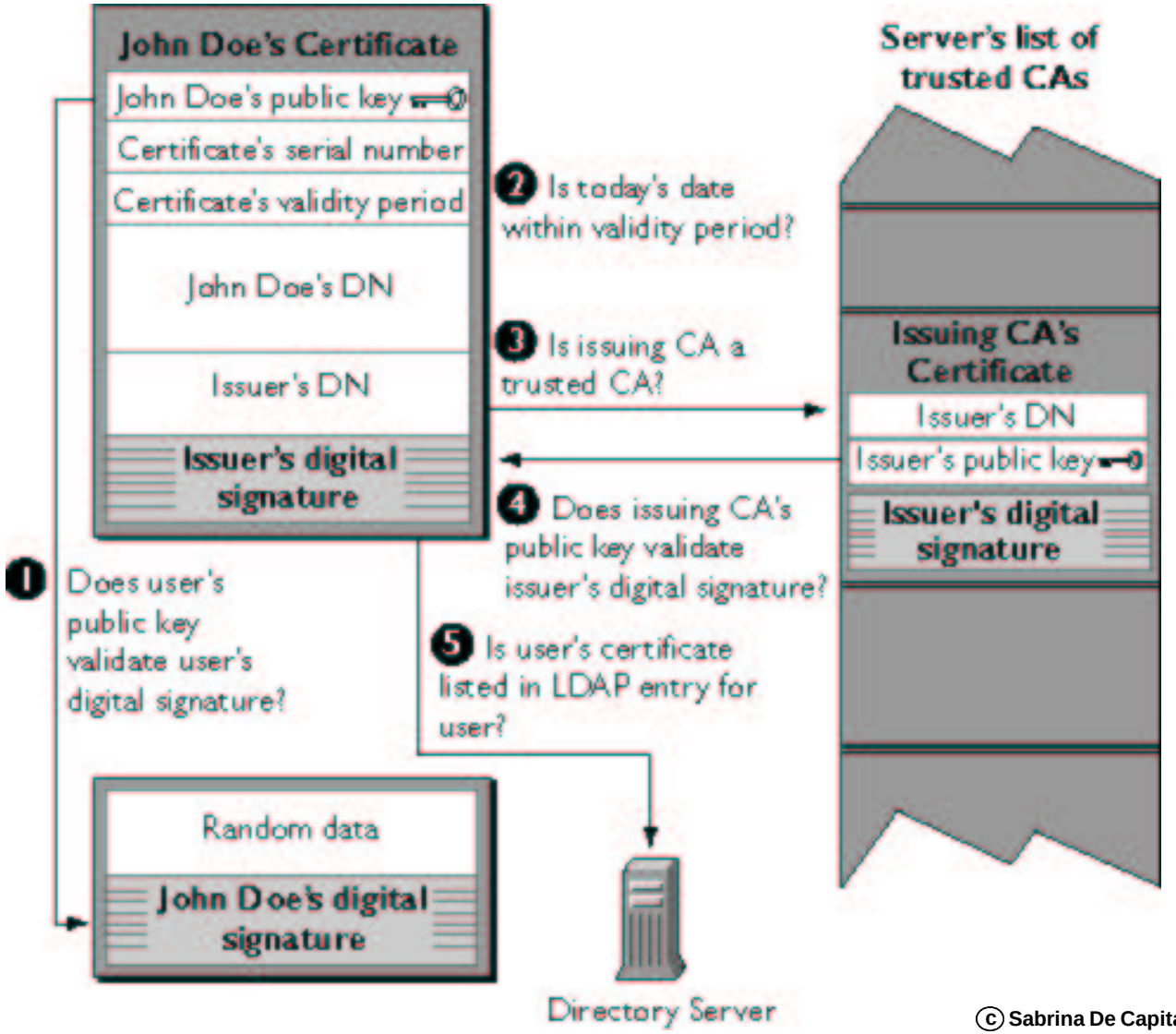
Server

- decritta il messaggio
- controlla l'integrità usando la funzione di hash
- risponde al client

Autenticazione Server



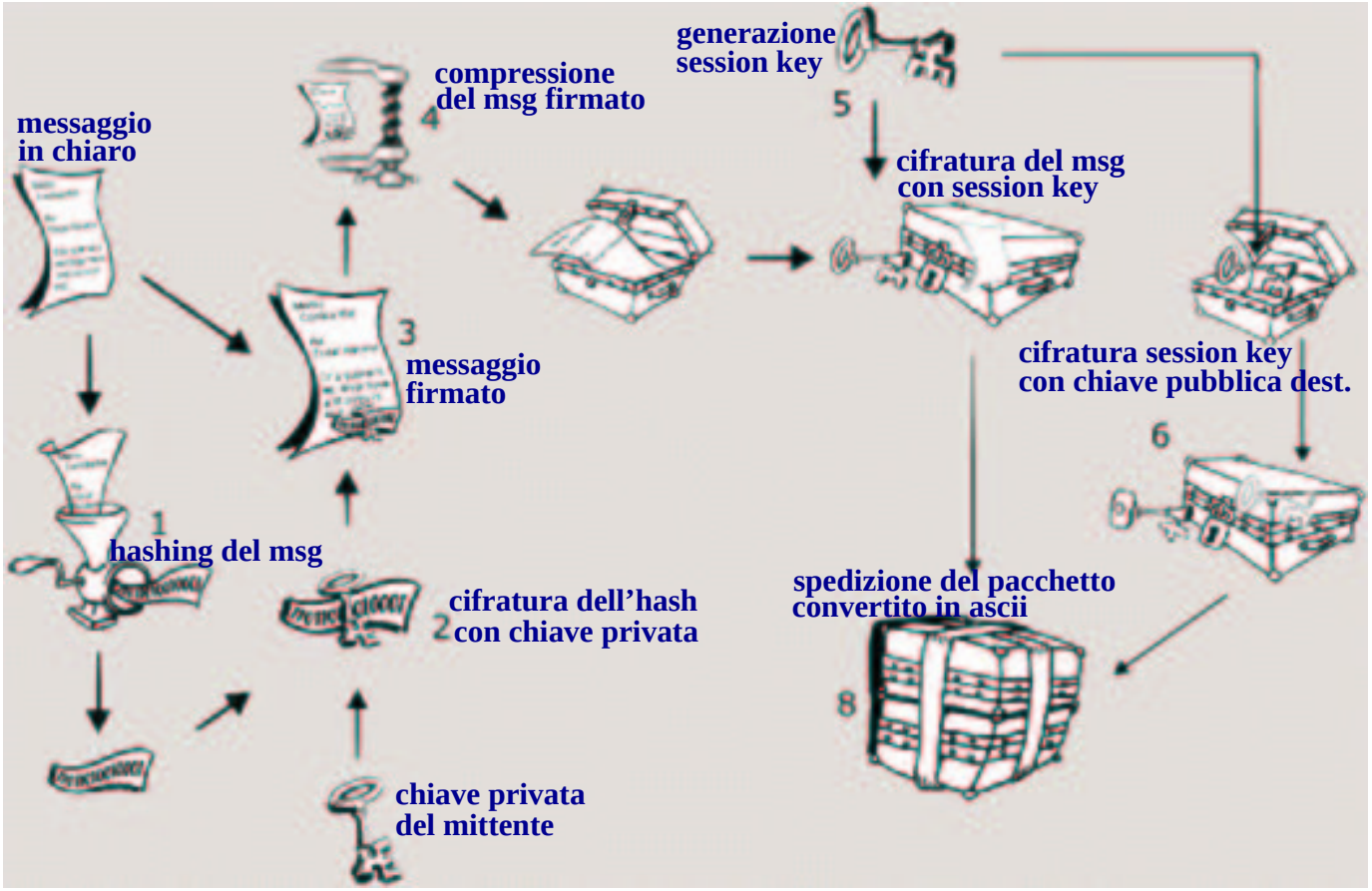
Autenticazione Client



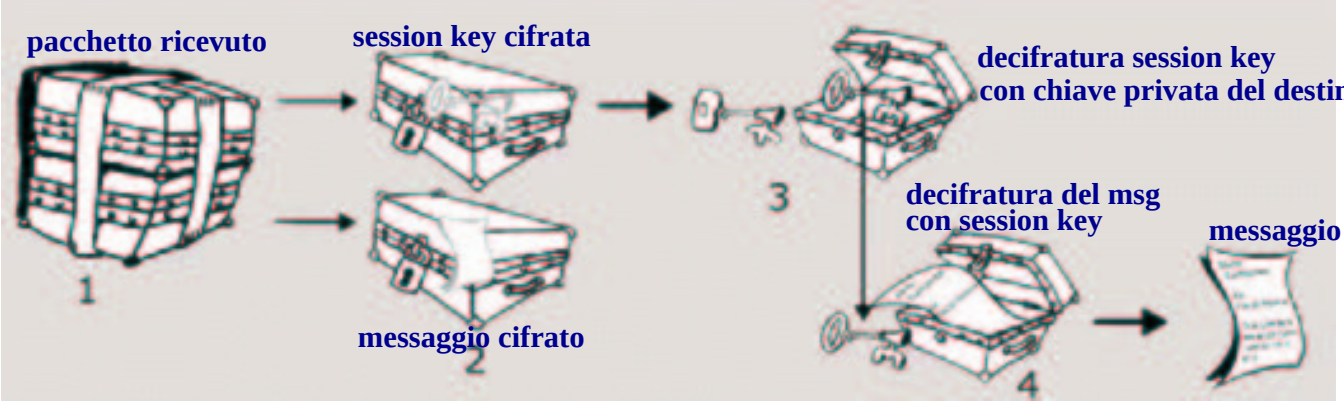
Pretty Good Privacy

- È un software di crittografia per la posta elettronica e la protezione di file di uso personale
- Creato da Philip Zimmermann nel 1991 e distribuito gratuitamente su Internet
- Cosa fa?
  - permette di firmare una e-mail lasciando il testo in chiaro
  - permette di cifrare una e-mail senza firmarla
  - permette di fare entrambe le cose

PGP: Procedura di Crittazione



PGP: Procedura di Decrittazione



Quali algoritmi vengono usati?

Funzione	Algoritmo	Descrizione
cifratura messaggio	CAST, IDEA, 3DES	messaggio cifrato con algoritmo simmetrico tramite session key
scambio s.k.	DH, RSA	session key è cifrata con chiave pubblica destinatario
firma digitale	RSA, MD5, SHA, DSS	creato codice hash del msg cifrato con chiave privata mittente
compressione	ZIP	msg compresso per la memorizzazione e trasmissione
compatibilità e-mail	Radix-64	msg cifrato è convertito in stringa ASCII per fornire trasparenza alle applicazioni e-mail
segmentazione	Radix-64	PGP esegue segmentazione e riassemblamento di msg di dimensione eccessiva

Protocollo Diffie-Hellman per Scambio di Chiavi

- Sfrutta le proprietà one-way della funzione del calcolo del *logaritmo discreto*
- Alice e Bob si scambiano “alla luce del sole” alcune parti della session key
- le parti scambiate non permettono ad un crittoanalista di risalire alla sessione key
- le parti scambiate combinate con le informazioni segrete conosciute da Alice e Bob (tali informazioni sono diverse) consentono di generare la stessa session key

Diffie-Hellman (1)

Esistono due elementi pubblici

1. q: numero primo
2. α: radice primitiva di $\mathcal{Z}_q^*$

Alice

- seleziona un intero $x_A < q$ (x_A chiave privata di Alice)
- genera $y_A = \alpha^{x_A} \mod q$ (y_A chiave pubblica di Alice)

Diffie-Hellman (2)

Bob

- seleziona un intero $x_B < q$ (x_B chiave privata di Bob)
- genera $y_B = \alpha^{x_B} \mod q$ (y_B chiave pubblica di Bob)

- Alice e Bob si scambiano le chiavi pubbliche y_A e y_B
- Alice calcola la session key $sk = y_B^{x_A} \mod q$
- Bob calcola la session key $sk = y_A^{x_B} \mod q$

Key Ring

Il PGP considera due strutture dati

- Private key ring usato per la memorizzare la coppia di chiavi, pubblica e privata, dell'utente, dove però la chiave privata è cifrata con una *passphrase* nota solo all'utente
- Public key ring usato per la memorizzare le chiavi pubbliche delle persone note all'utente

Campi del Private Key Ring

- timestamp: ora in cui è stata generata o inserita una chiave
- keyid: 64 bit meno significativi della chiave pubblica
- public key: chiave pubblica
- private key: chiave privata cifrata
- user id: il proprietario della chiave

Campi del Public Key Ring

- timestamp: ora in cui è stata generata o inserita una chiave
- keyid: 64 bit meno significativi della chiave pubblica
- public key: chiave pubblica
- owner trust: fiducia nel proprietario della chiave
- user id: il proprietario della chiave
- key legitimacy: fiducia nella chiave
- firma: firma per la chiave
- signature trust: fiducia nella firma

Chiavi Pubbliche e Keyserver

- La chiave pubblica si ottiene direttamente dalla persona interessata o si possono usare i keyserver
- Sono particolari server presenti su Internet dedicati al deposito e prelievo delle chiavi pubbliche, in rete tra loro, per cui una chiave immessa in un server viene diffusa anche sugli altri
- Per ricevere o inserire chiavi bisogna inviare una e-mail all'indirizzo del keyserver

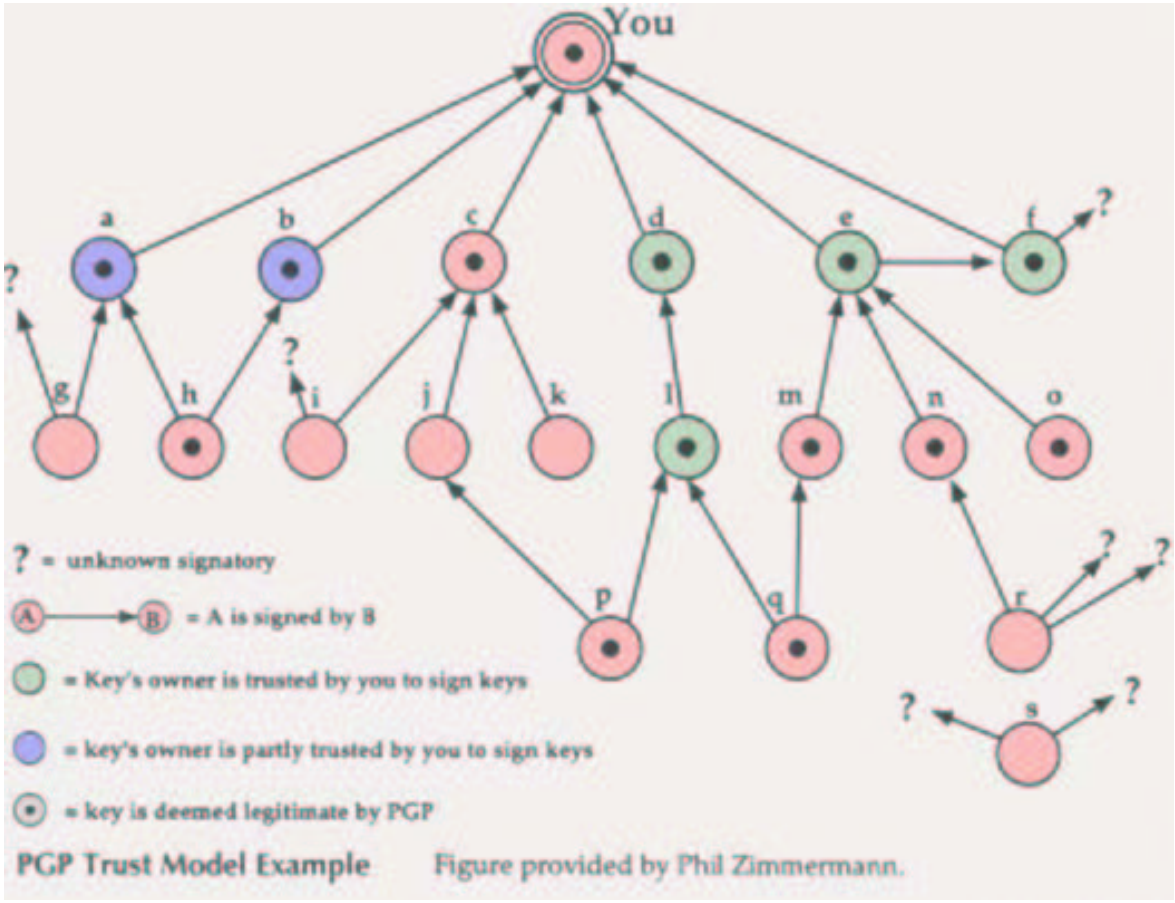
Web of Trust: Cosa Significa?

- PGP si basa su una gestione decentralizzata, in cui ciascuno si rende responsabile certificando una firma o un'altra
- Non bisognerebbe mai certificare una chiave di cui no si è perfettamente sicuri perché se fosse fasulla si comprometterebbe il "web of trust" che PGP mira a creare

Livelli di Trust

- Ci sono quattro livelli di fiducia che si possono assegnare alla chiave pubblica in un key ring
  - completamente fidato
  - parzialmente fidato
  - non fidato
  - non noto
- Tutte le chiavi firmate con la propria chiave sono valide

Web of Trust: Un esempio....



Dove Trovare PGP

Tutte le versione di PGPi si possono trovare all'indirizzo: www.pgpi.com

Alla luce della nuova legislatura sarebbe possibile scaricare la versione PGP6.x dal MIT distribution site for PGP all'indirizzo: www.mit.edu/network/pgp.htm.