

Comunicazioni Sicure

Esempio: fare connessione con 2 terminali e farli parlare tra di loro.

Corso di reti di
Calcolatori e
Cybersecurity

<date>
Location



Agenda

→ Come faccio a sapere che chi parla è chi dichiara di essere?

- Autenticare Chi partecipa ad una comunicazione
- Scambio di messaggi Sicuri Informazioni che non devono essere alterate
- I Certificati e le infrastrutture a Chiave Pubblica
- La Firma Digitale
- SSL

AUTENTICAZIONE

Autenticazione

- “The process of verifying an identity claimed by or for a system entity.” RFC 2828
- Il processo è composto da:
 - **Identification**: La fase nella quale viene specificato l’identificatore: quando tu dichiari chi sei. Associazione la persona al modo in cui è associata al sistema (utente)
 - **verification** – Associa L’identificatore all’identità della persona. \Rightarrow Verificare che l’associazione è corretta. Username e password

Verifico che qualcuno è chi dichiara di essere

\neq

I 4 Principi per l'Autenticazione Utente

→ Autenticazione a 2 fattori

- **Something the user knows**

- Esempio: password ...

- **Something the user has**

- Esempio: smartcards ...

↳ mod da adottare

- **Something the user is**

- Esempio: Impronta digitale ..

- **Something the user does**

- Esempio: l'esecuzione della firma..

PIN + bancomat è autenticazione a 2 fattori (PIN + possesso carta)

Protocolli di Autenticazione

- Protocollo dedicato ad autenticare un utente attraverso la rete
- Problemi di sicurezza:
 - Spoofing (falsificazione dell'identità)
 - Spacciarsi per un altro utente/sistema Rubare identità
 - Eavesdropping/Sniffing
 - Intercettazione dei pacchetti di rete
 - Effetti: furto di password,
 - Tampering
 - Alterazione dei pacchetti
 - replica di una sequenza intercettata, ...

Problema doppio: l'utente che chiude la password, la rete di comunicazione e l'utente effettivo.
Se io sono ora remoto a chiedere, si aggiunge problema di furto pw da rete, alterare pw dalla rete, impedire comunicazione rete

ES. Attacco a integratori cambiando pacchetti, attacco a disponibilità fisica per sempre sfuggente,
ridurre per: attacco a confidenzialità.

Schema di Autenticazione diretto

Tecniche più usate. Sono schemi
(password, non protocollo).

Azioni e Dati Client	Dati Scambiati	Azioni e Dati Server
	←Auth REQ	
U: userID P: Password AUTH={U,P}	AUTH → <i>mando pacchetto di U e Pw in un solo pacchetto ols credenziali</i>	
	←REPLY	P(U): password di U nel Server if P(U)=P REPLY=yes else REPLY=no

Mecanismo é sbagliato; Problem:

- 1) AUTH può essere inviolabile e letto, e ho tutte le info. Debba a sniffing.
- 2) Mecanismo prevede che server agisca per protocollo. Ma rapporto client-server non è lo stesso fra tutti i livelli: es. server su livello applicativo può anche essere client su TCP.

PROTOCOLLO: individuare punti di graco
tipi di message scambiati
formato dei message
regole di interscambio.

HTTP prevede solo request e reply. Client ha solo request, server solo reply.
Soluzione è difficile per intercettare AUTH. 1 modo è dare un assodato che so chi è client
e server, ma comunque non è sicuro. Es. la chiamé può essere modificata posteriormente

2) Reply è unico. Io prendo pacchetto sniffato, non so cosa c'è scritto e lo rimando.

Autenticazione Remota basato su sfida

Modello servizio del tipo: ti spiego a dimostrazione chi devi di essere.

Azioni e Dati Client	Dati Scambiati	Azioni e Dati Server
	←Auth REQ	
U: userID <i>rispondo con Username</i>	U→	Store U, Challenge
	←CHALLENGE	
P: Password PrepareReply(P)	Proof(P,U,Challenge)→ <i>Ti do proof che dipende da P, U e challenge.</i>	P(U): password di U nel Server if Proof(P,U,Challenge) is ok REPLY=yes else REPLY=no

*Cosa bancherei ancora quello di prima, altrettanto la proof è più complessa.
Parte sempre col server che controlla più.*

Schema di Autenticazione basato su sfida con hash

Azioni e Dati Client	Dati Scambiati	Azioni e Dati Server
	←Auth REQ	
U: userID	U→	Store U, Challenge
	←CHALLENGE	$r = \text{genera numero random}$ $h(), f()$ funzioni hash $\text{CHALLENGE} = \{r, h(), f()\}$ Store r per questa session
P: Password R: $f(r, h(P))$	R → ↳ Anche se ho R non sono in grado di riconoscere $h(P)$. ←REPLY	↓ Sequenza formata da numero casuale e funzione di hash. P(U): password di U nel Server if $R = f(r, h(P(U)))$ REPLY=yes else REPLY=no

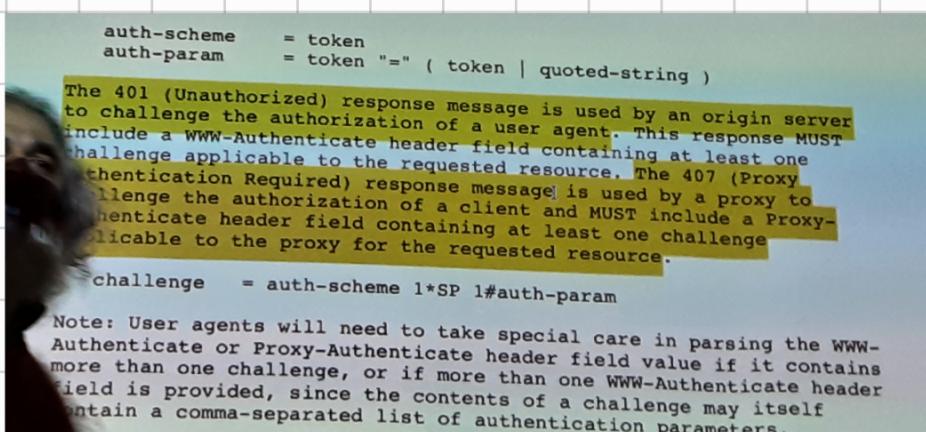
Una usata per le verifiche, una per hash password
Password mai mandata.

Nel server mette hash delle password con salt che concordano alla password per avere hash diverso su pw uguali.

Salt per utente rimane sempre lo stesso perché è salvato in locale l'hash.

DIGRESSIONE RFC

- Cognitivo da numero e working group.
- Abstract porta sintesi di informazioni.
- Se mando richiesta a risorsa autorizzata, lo standard manda errore 401 (che spesso non richiede), il browser apre la finestra e manda richiesta. Se faccio cancel esce errore 401.
Il serv. manda stringa WWW-Authenticate header field.



Parametri sono:

realm = identifico contesto da cui ci autorizzo. Ogni risorsa corrente si trova in un realm differente. Io invendo un messaggio di request con header di autorizzazione.
In questo modo io mando direttamente questa richiesta senza bisogno di errore.

2. Basic authentication scheme: meccanismo di challenge è fatto da "Basic" e uno parametro di realm.

Come credenziali ho "Basic" e basic credentials.

RFC 2617: Autenticazione con HTTP

This document also provides the specification for HTTP's authentication framework, the original Basic authentication scheme and a scheme based on cryptographic hashes, referred to as "Digest Access Authentication"

Funzione di Calcolo Digest

- $A1 = \text{username:realm:passwd}$
- $HA1 = \text{MD5}(A1)$
- $A2 = \text{method:digestURI}$
- $HA2 = \text{MD5}(A2)$
- $\text{Response} = \text{MD5}(HA1:\text{nonce}:HA2)$

↑
così mi proteggo da forging
Non sia possibile aggiungere dati
al nonce. Length extention protection

HTTP Authentication framework

- NON e' un meccanismo di autenticazione particolarmente "forte"
 - Basic: username e Pwd praticamente in chiaro (*sconsigliata*)
 - Digest: basata su challenge, usa funzioni hash (*più probabile*)

HTTP Authentication framework

- NON e' un meccanismo di autenticazione particolarmente "forte"
 - Basic: username e Pwd praticamente in chiaro
 - Digest: basata su challenge, usa funzioni hash
- Domanda :
 - Come misurare la forza di un sistema di autenticazione?

HTTP Authentication framework

- NON e' un meccanismo di autenticazione particolarmente "forte"
 - Basic: username e Pwd praticamente in chiaro
 - Digest: basata su challenge, usa funzioni hash
- Domanda :
 - Come misurare la forza di un sistema di autenticazione?
 - **Risposta:** Misurare quantitativamente la sicurezza è necessario , ma non semplice, al momento pochi risultati concreti. La forza dipende dalla verifica della resistenza agli attacchi noti.... *Temi da approfondire in una fase successiva della preparazione*

Chiamare dunque un numero per dire che una cosa è più sicura dell'altra sta sognando certe.
Ma sicurezza è un problema. Che significa dire che una cosa due è più sicuro? Una si rompe
e l'altra no?

- Questo ci vuole per arrivare a complessità sufficiente per rompere quell'algoritmo?

Apache e Basic Authentication

ESERCITAZIONE

Autenticazione Basic in una Applicazione NodeJS

- Sviluppare una applicazione NodeJS che effettui la HTTP Basic authentication

Vedere Esempio di
Codice

Installazione e Configurazione Apache

- Scaricare ultima versione di Apache Web Server
- (Installazione cambia da utente a utente)
- Configurare Virtual Host
 - Nome host in /etc/hosts
 - Abilitare VirtualHost in httpd.conf
 - Include <path>/apache/extrahttpd-vhosts.conf

PROVA A FARLO

```
<VirtualHost *:80>
    DocumentRoot /Applications/MAMP/htdocs
    ServerName localhost
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "/Users/username/Sites/mysite"
    ServerName mysite.loc
</VirtualHost>
```

- Crea file index.html

- Configurare Apache per permettere autenticazione
 - Cercare la riga: <Directory /> AllowOverride AuthConfig
- Crea file .htaccess nella home del virtual host
 - Configurare

```
AuthType Basic  
AuthName "Secure Content"  
AuthUserFile /home/myuser/public_html/.htpasswd  
require valid-user
```
- Creare il file degli utenti
 - Nella cartella home eseguire:

```
# htpasswd -c /home/myuser/public_html/.htpasswd myuser
```
- Fare il restart di Apache

Memorizza nel file pw + salt hashati.

Configurare Autenticazione Basic

- Crea file .htaccess nella home del virtual host
 - Configurare

```
AuthType Basic  
AuthName "Secure Content"  
AuthUserFile /home/myuser/public_html/.htpasswd  
require valid-user
```

- Creare il file degli utenti
 - Nella cartella home eseguire:

```
# htpasswd -c /home/myuser/public_html/.htpasswd myuser
```

- Configurare Apache per permettere autenticazione

- Cercare la riga: <Directory />

```
AllowOverride AuthConfig
```

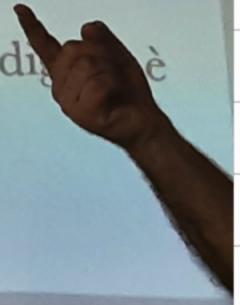
- Fare il restart di Apache

Intercettare il Messaggio di Autenticazione

- Avviare wireshark
- Accedere alla pagina
- Effettuare l'autenticazione
- Fermare Wireshark
- Identificare lo scambio di messaggi HTTP
- Trovare il messaggio con Header “Authorization”
- Recuperare le credenziali Basic
- Decodificare (base64) le credenziali

Configurare Autenticazione Digest

- (La configurazione in httpd.conf è la stessa di basic)
- Crea file .htaccess nella home del virtual host
 - Configurare
 - AuthType: Digest
 - Cambiare il file passwd
- Creare il file degli utenti
 - Nella cartella home eseguire:
 - Htdigest -c <file> <realm> <user>
- Il formato dei file htpasswd (basic) e htdigest (digest) è diverso ..

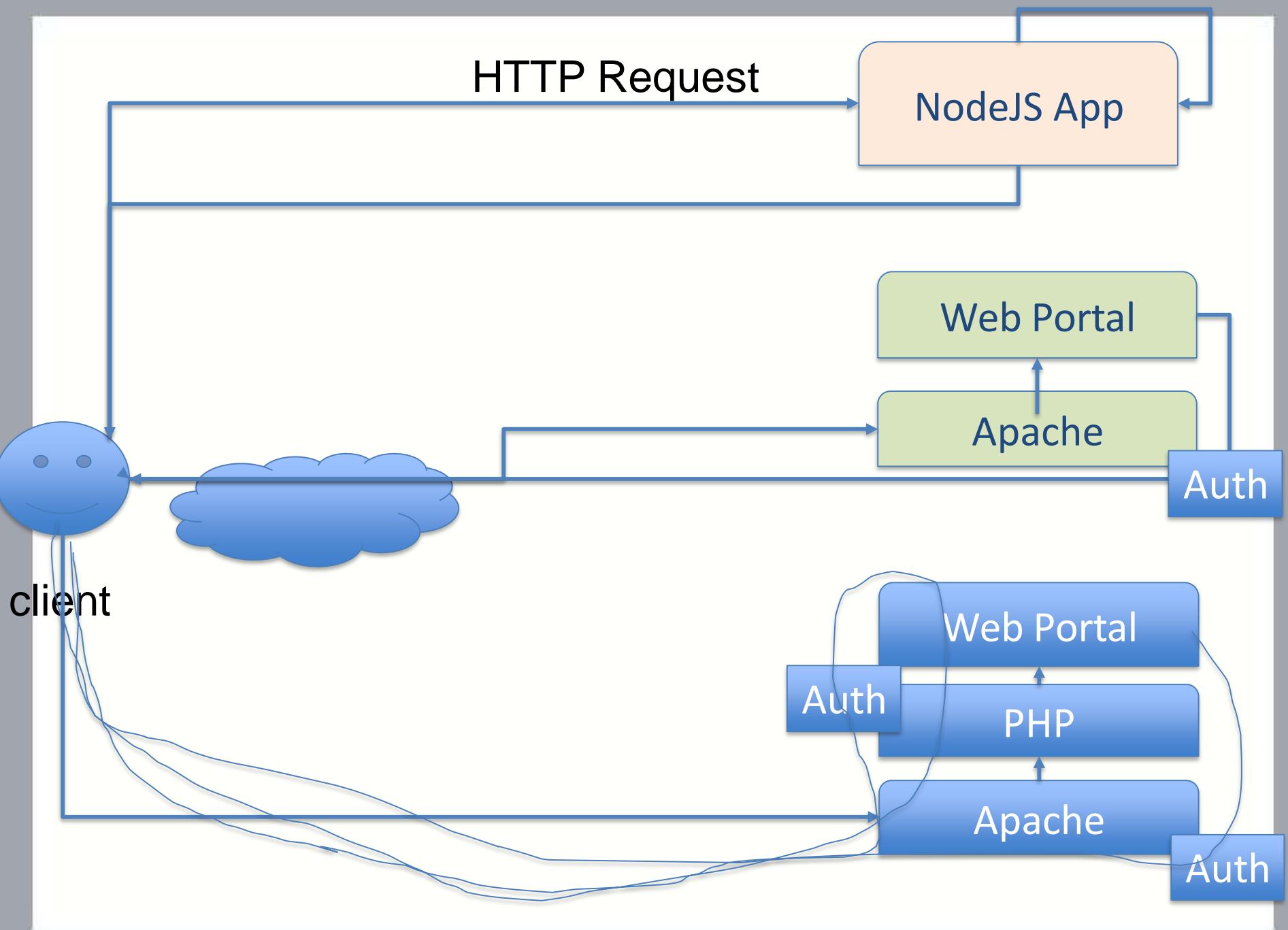


Intercettare il Messaggio di Autenticazione

- Avviare wireshark
- Accedere alla pagina
- Effettuare l'autenticazione
- Fermare Wireshark
- Identificare lo scambio di messaggi HTTP
- Trovare il messaggio con Header "WWW-Authenticate"
- Trovare il messaggio con Header "Authorization"
- Recuperare le credenziali Digest
- Provare a ricalcolare il Digest

Considerazioni

- Quale è la differenza nei due approcci?
- Che problemi di sicurezza si pongono in entrambi i casi?



**SCAMBI SICURI CON
CRITTOGRAFIA SIMMETRICA**

Il Problema di Sicurezza

Descrizione del Problema

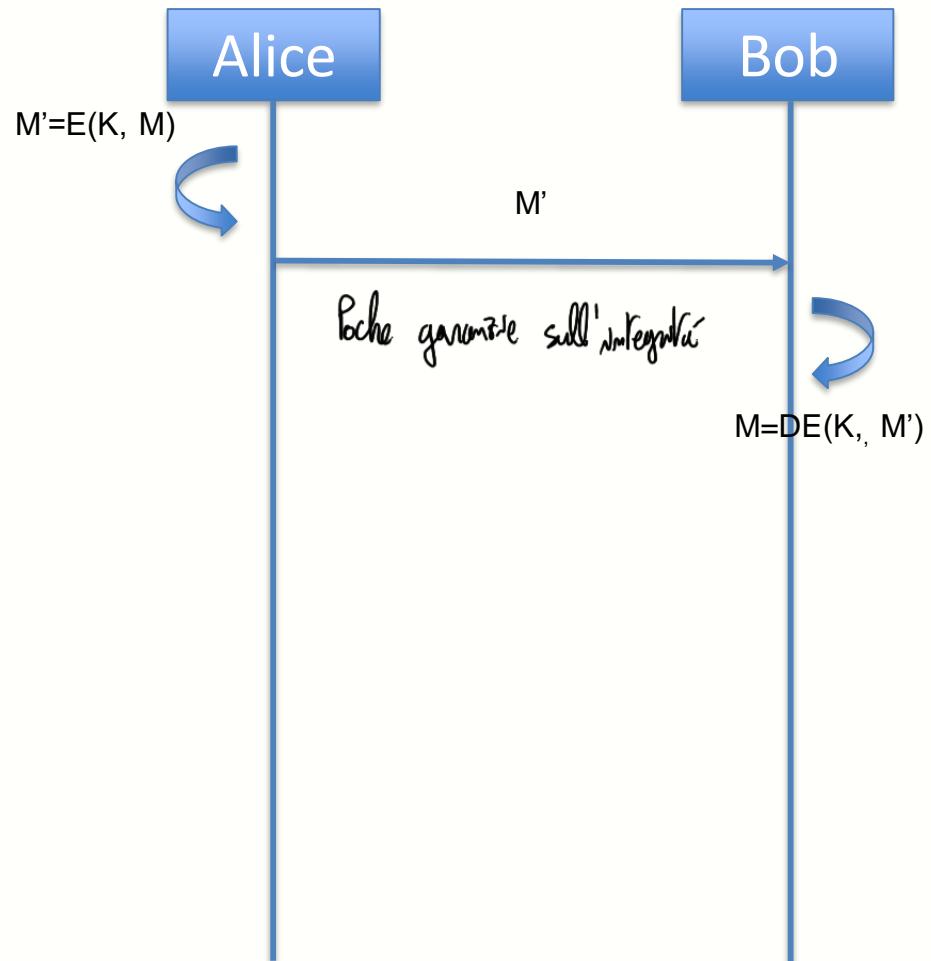
- Bob e Alice vogliono scambiarsi dei messaggi
- I messaggi riguardano temi privati e non vogliono che altri possano leggere il contenuto
- Inoltre hanno paura che qualcuno possa alterare i messaggi, falsando la comunicazione

Formalizzazione

- Attori coinvolti:
 - Bob e Alice devono scambiare un messaggio
- Requisiti:
 - Confidenzialità
 - Integrità

Soluzione del Problema

- Alice e Bob si accordano (off-line) per una chiave segreta condivisa
- I messaggi vengono crittati con la chiave segreta



Considerazioni

- La comunicazione è sicura fino a quando:
 1. L'algoritmo adottato è sicuro
 2. La chiave è mantenuta segreta
- Adottare Best Practice per
 - Scelta e Implementazione degli algoritmi
 - Gestione delle Chiavi
- Una soluzione comune è utilizzare la crittografia asimmetrica per scambiare la chiave da usare per la crittografia simmetrica...

**SCAMBI SICURI CON
CRITTOGRAFIA ASIMMETRICA**

Il Problema di Sicurezza

Descrizione del Problema

- Bob e Alice vogliono scambiarsi dei messaggi
- I messaggi riguardano temi privati e non vogliono che altri possano leggere il contenuto
- Inoltre hanno paura che qualcuno possa alterare i messaggi, falsando la comunicazione

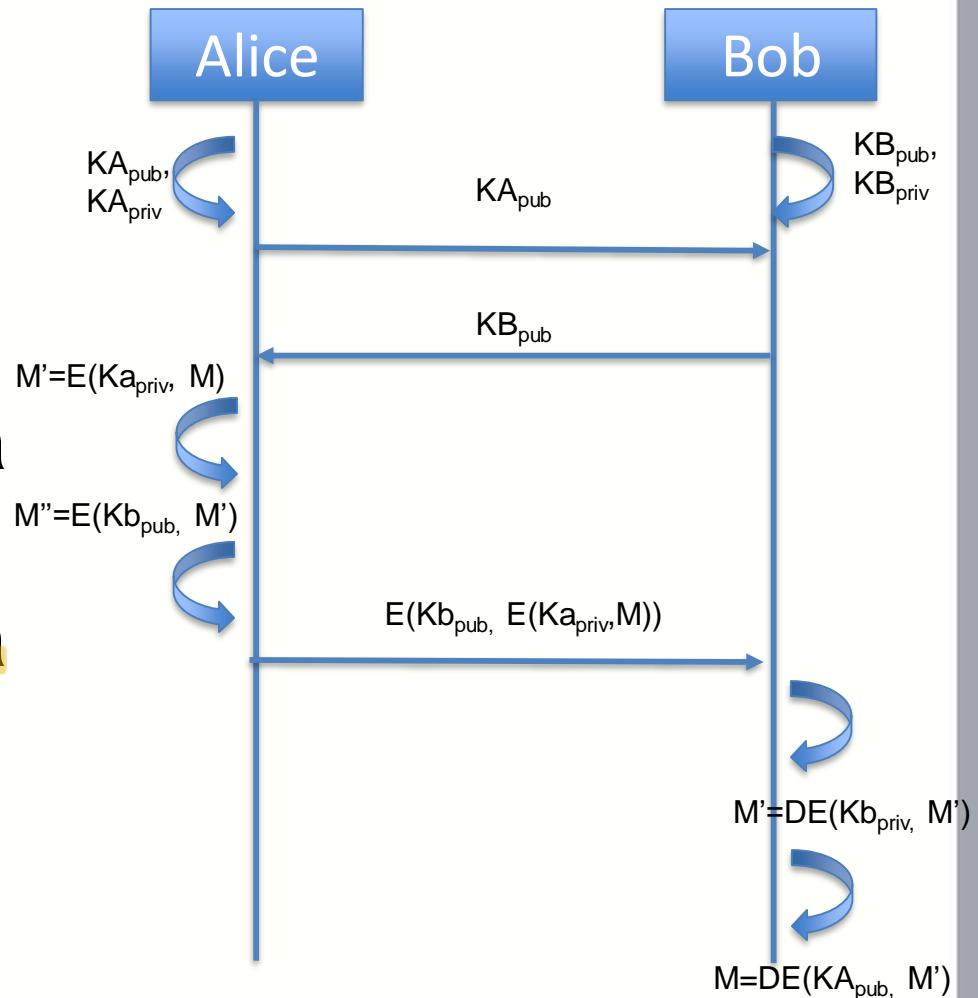
Formalizzazione

- Attori coinvolti:
 - Bob e Alice devono scambiare un messaggio
- Requisiti:
 - Confidenzialità
 - Integrità

Soluzione del Problema

Se i messaggi sono di dimensioni corrette (né piccole né grandi), avendo doppia crittazione aumenta la sicurezza.

- Alice e Bob generano una coppia di chiavi con crittografia asimmetria
- Alice fornisce a Bob la sua chiave pubblica
- Bob fornisce ad Alice la sua chiave pubblica
- I messaggi vengono crittati ogni volta con entrambe le chiavi



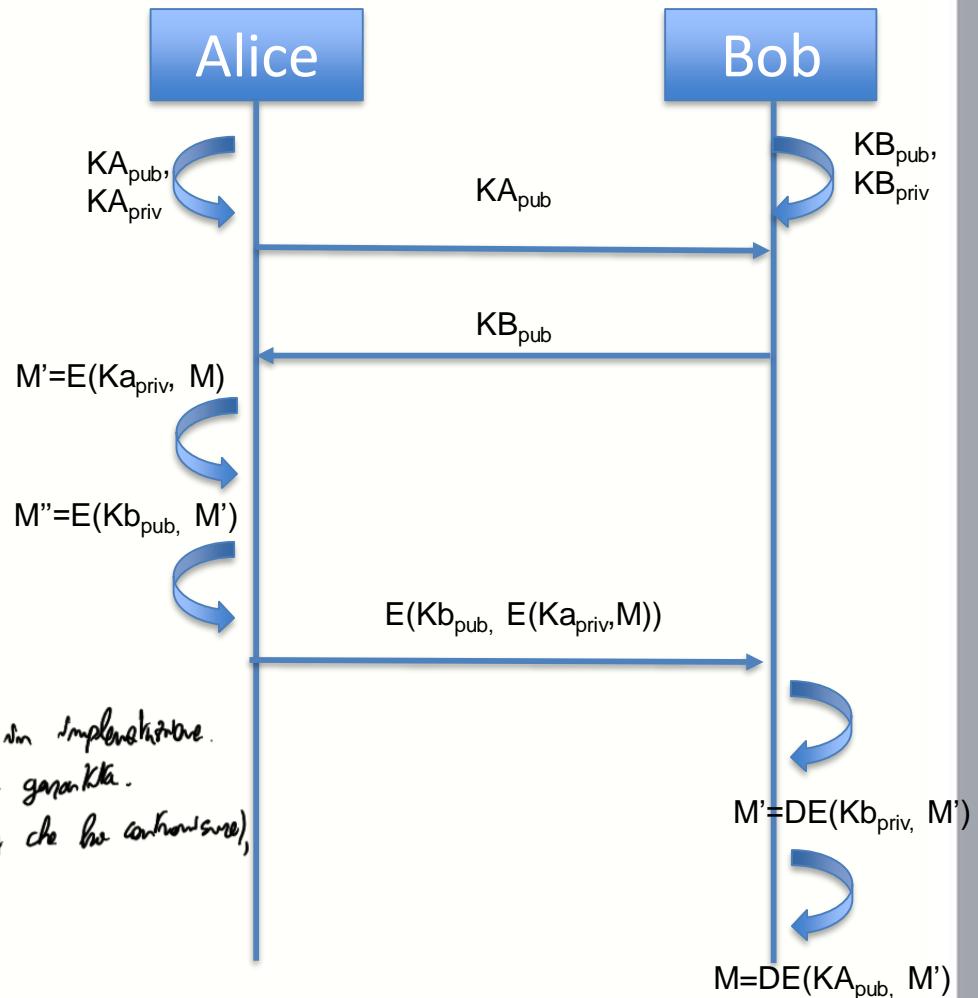
Soluzione del Problema

- Questa Soluzione è vulnerabile.
- Dove?

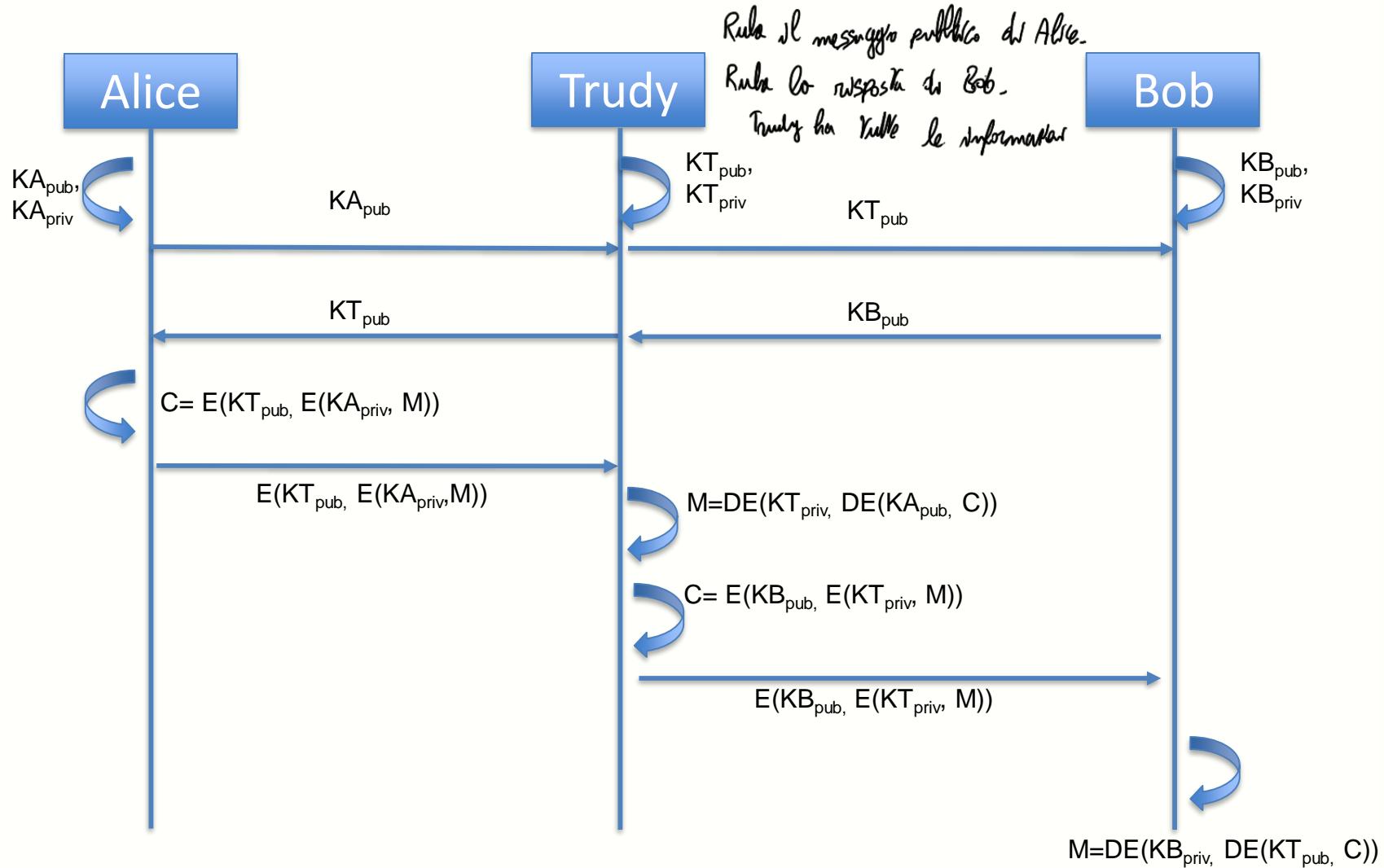
Chi ha detto che la chiave pubblica
vive in Alice?

Non ho garanzie su se che dicono
Trudy cambia sp di Alice, ma Bob risponde
ad Alice, per fare un retroscena di
questo tipo (su una rete locale) dovre usare ARP
poisong.

1. Protocollo matematicamente sicuro non è detto che sia corretto in implementazione.
Assunzione di provabilità, amaro e man allorabilità dove essere garantita.
2. Altracce segnale: sulla cavia è facile (in locale ARP poisong che ha conoscenze),
Su rete globale è difficile.



Attacco Man-in-the-Middle



Hp: Trudy riceve messaggi ma non può impedire l'arrivo. Invia entrambi.

Bob risponde inviando la sua chiave e Alice riceve entrambe le chiavi (Bob e Trudy).

Se Bob riceve messaggio nuovo da Alice, dovrà gestire 2 messaggi che gli arrivano.

NOTA: sistemi di autenticazione logga tutti i tentativi di accesso per valutare accesso.

E come mi difendo da DDoS? Dipende dalle policy.

Man-in-the-Middle Attack

1. Trudy prepares by creating two private / public keys
2. Alice transmits her public key to Bob
3. Trudy intercepts this and transmits his first public key to Bob. Trudy also calculates a shared key with Alice
4. Bob receives the public key and calculates the shared key (with Trudy instead of Alice)
5. Bob transmits his public key to Alice
6. Trudy intercepts this and transmits his second public key to Alice. Trudy calculates a shared key with Bob
7. Alice receives the key and calculates the shared key (with Trudy instead of Bob)
8. Trudy can then intercept, decrypt, re-encrypt, forward all messages between Alice & Bob

Conclusioni

Problema è più le implementazioni

- Il Punto debole NON è la crittografia, ma lo scambio di chiavi
- Quando vengono scambiate le chiavi pubbliche non c'è nessuno che garantisca l'identità di chi le ha fornite.
- Le Infrastrutture a Chiave Pubblica, servono a fare questo: Una Terza Parte fidata che garantisca l'identità

Un tipico Problema di Sicurezza

FIRMA DIGITALE

Firma Elettronica e Firma Digitale

Firma Elettronica e Firma digitale non hanno lo stesso significato, in particolare dal punto di vista legislativo

*The term ‘‘electronic signature’’ means an electronic sound, symbol, or process, attached to or logically associated with a contract or other record and executed or adopted by a person with the intent to sign the record
(Electronic Signatures in Global and National Commerce Act, E-Sign)*

(Il termine “firma elettronica” indica un qualsiasi simbolo, suono o processo elettronico collegato o logicamente associato ad un contratto o altro documento, eseguito con l’intento di firmare il documento stesso)

Il termine **firma elettronica**, quindi, come indentificato dalla legge US, non implica obbligatoriamente l’utilizzo di tecniche crittografiche per garantire identità, integrità o altre proprietà di sicurezza

Obiettivo: garantire che documento è legittimo e non riproducibile
Ma la firma scritta è riproducibile! Come fissa?

Firma Digitale

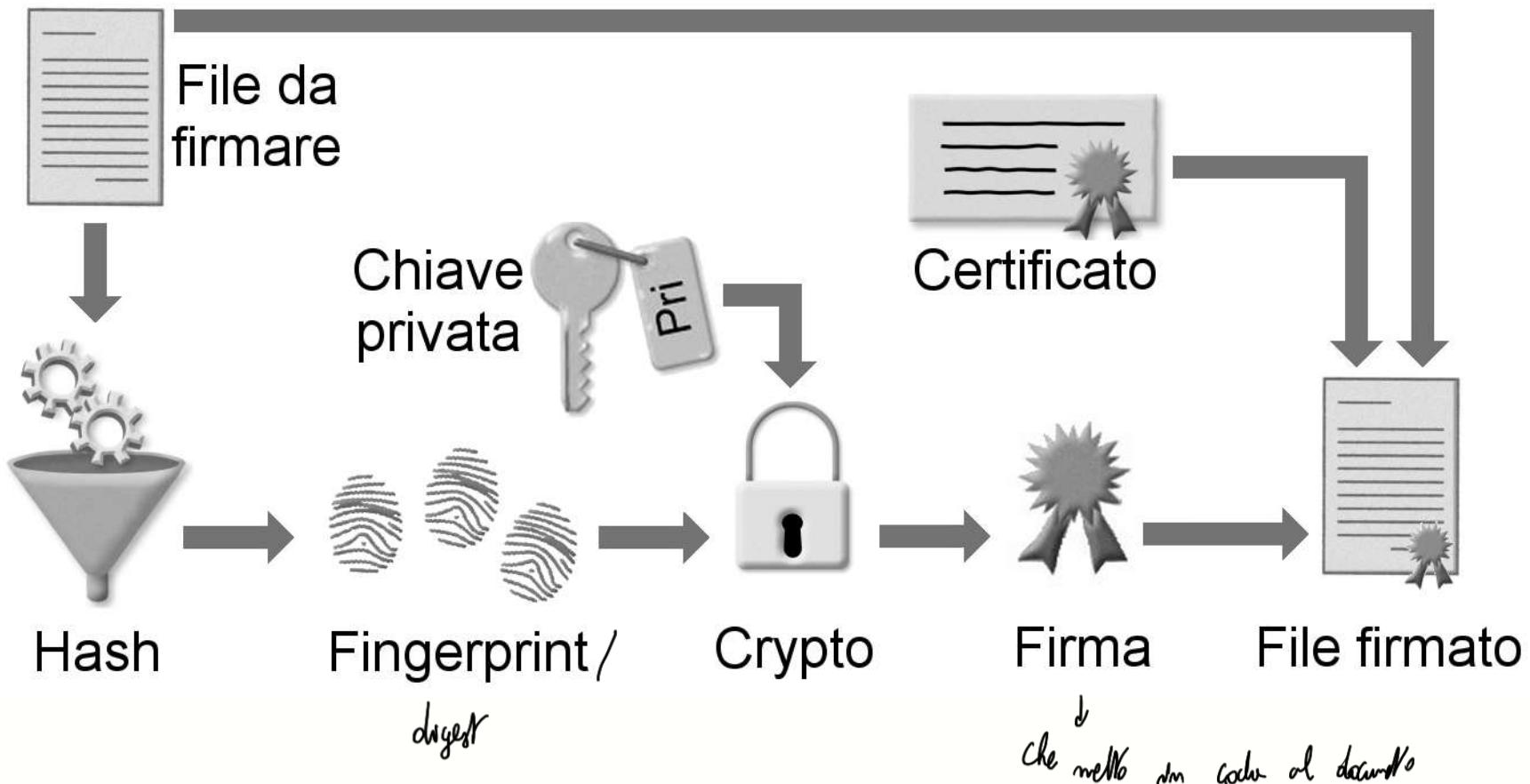
E' un particolare tipo di firma elettronica, che combina funzioni hash con la crittografia asimmetrica

- Utilizzo della funzione hash per generare un *digest* di dimensioni fissate: genera hash del documento, che garantisce integrità ma non sorgente.
- Utilizzo della crittografia asimmetrica per garantire integrità della firma (con hash con chiave privata e documenti con chiave pubblica).
- Utilizzo di certificati per garantire l'identità (anche chiave con certificato che garantisce che questo proviene dalla persona giusta. \Rightarrow Firma digitale si divide in qualificata e non qualificata a seconda del provvedere autoritativo).

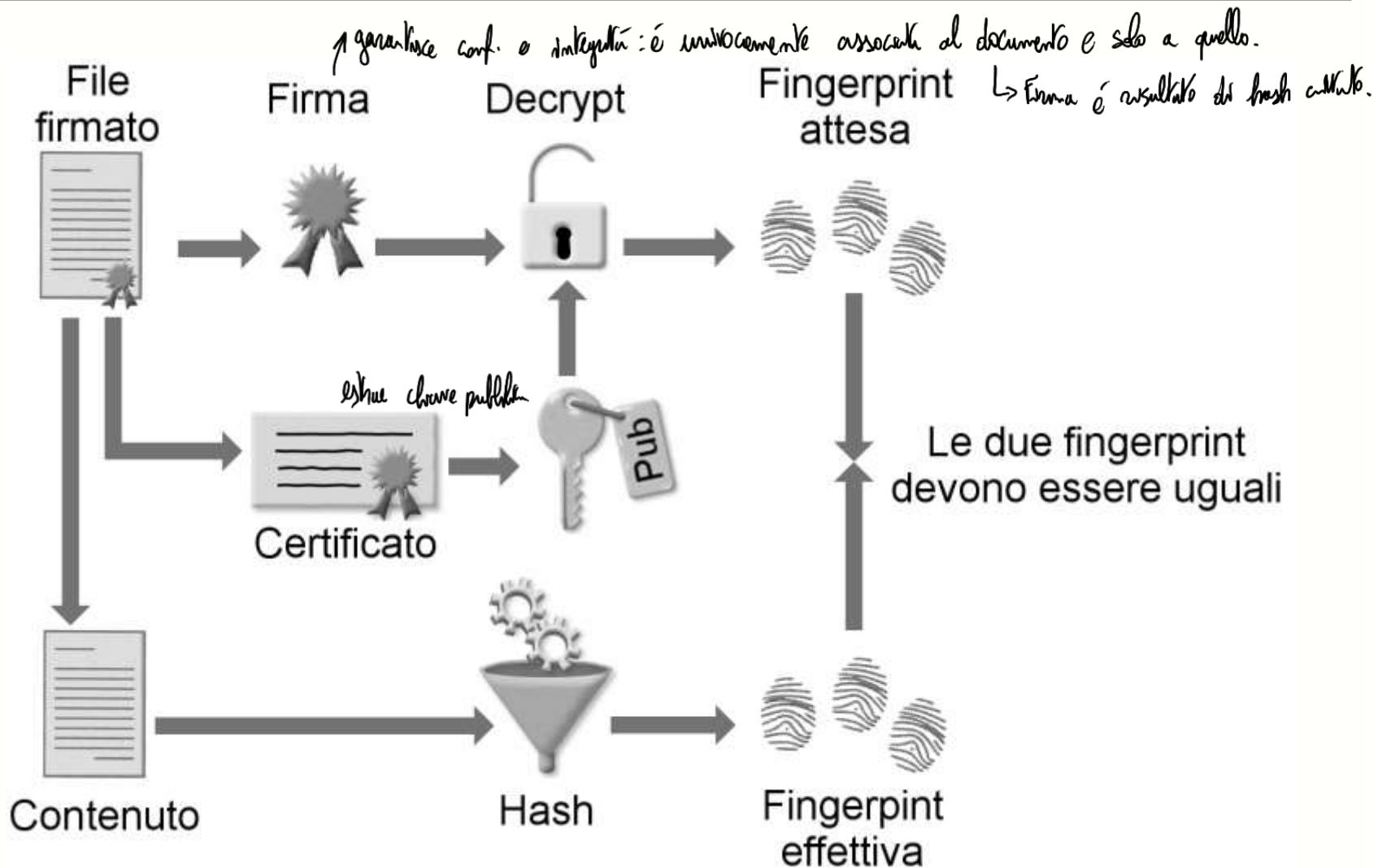
Caratteristiche:

- Integrità
- Non ripudiabilità [anche solo della persona. Istituzioni con certificato]
- NON GARANTISCE CONFIDENZIALITÀ'

Signature production



Signature validation



Problema, certificato deve essere formato.

Nota: certificati sono firmati.

INFRASTRUTTURE A CHIAVE PUBBLICA (PKI)

Infrastrutture a Chiave Pubblica

Le **Public Key Infrastructure (PKI)** hanno il **compito** di:

- Collegare le chiavi con l'identità dei proprietari
- Distribuire le chiavi “pubbliche”

(Definizione NIST)

Secondo IETF PKIX working group:

Una PKI è: “*The set of hardware, software, people, policies and procedures needed to create, manage, store, distribute, and revoke Public Key Certificates based on public-key cryptography*”

L'**insieme** di hardware, software, persone, politiche di sicurezza e procedure necessarie per **creare, gestire, mantenere, distribuire e revocare** certificati basati su chiave pubblica

Public Key Infrastructures

Public Key Infrastructure (PKI) provides the means to bind public keys to their owners and helps in the distribution of reliable public keys in large heterogeneous networks.

NIST

The set of hardware, software, people, policies and procedures needed to create, manage, store, distribute, and revoke Public Key Certificates based on public-key cryptography.

IETF PKIX working group

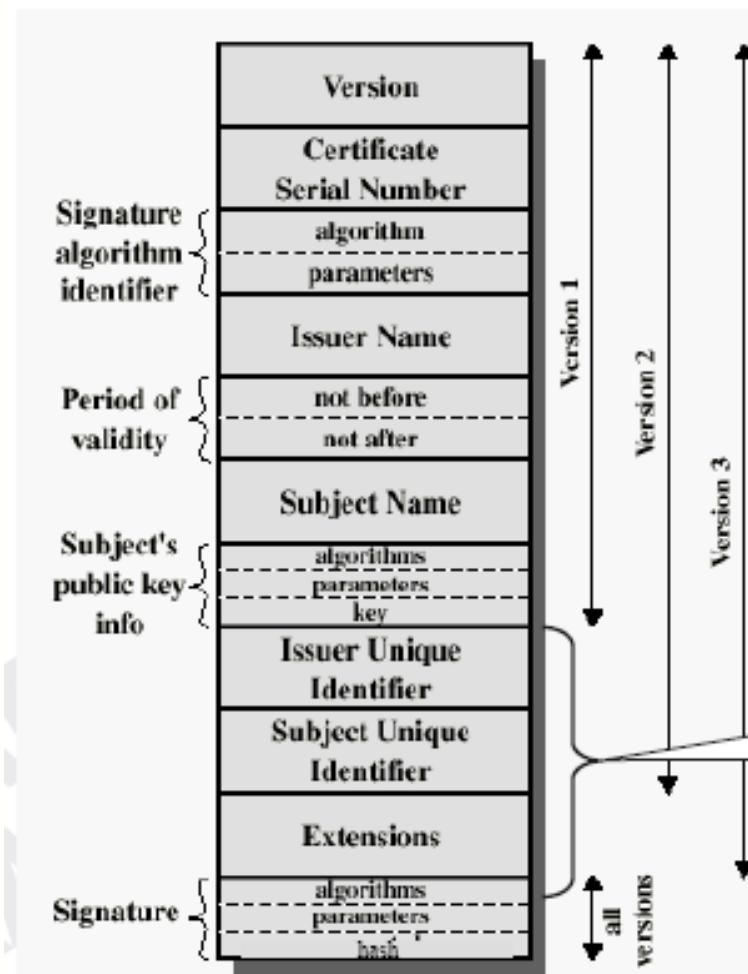
Distribuzione delle chiavi

- Certificato e chiave pubblica:
 - Documento firmato che specifica la chiave pubblica e l'identità del suo proprietario.
- Certificate Authority (CA)
 - Una agenzia responsabile di produrre e firmare i certificati
 - Dopo aver generato una coppia di chiavi, l'utente dimostra la sua identità presso l'agenzia (attraverso una Registration Authority – RA)
 - Il certificato attestante l'identità viene firmato dalla CA e reso pubblico

1. Come sono fatti questi documenti?
2. Come si sviluppano e si creano?

*Digital Certificates x.509 v3

RFC 2459,
RFC 3280



m. Vorone, sentile che relativamente
ai modi diversi di chi per
cifrare come vengono costituiti i
documenti, nome chi chi ha prodotto
certificato, nome chi chi ha
identikit associato a certificato
(Issuer Name), chiave (in genere
quella attuale), nome del soggetto,
info sulla chiave pubblica, id unico
di riferimento e soggetto, estensione
per la forma del certificato fatta
con la chiave privata dell'issuer
che rilascia documento (quindi non
intercettabile e non riproducibile)
Risulta ora da fine middle: "questa
chiave è propria di Alice".

Ho doc non alterabile con garanzia (da issuer) su identikit delle persone

x.509 fields

- **Version:** v1, v2, or v3.
- **Serial #:** a unique number.
- **Signature method:** The method used to sign the digital certificate (e.g., RSA).
- **Issuer name:** The entity whose private key signed the certificate.
- **Valid time period:** begin time and end time.
- **Subject name:** The entity whose public key is included in the certificate.
- **Subject's public key:** public key and public key method.

x.509 fields

Version	2 (V1=0, V2=1, V3=2)
Serial Number	56
Signature Algorithm	sh1RSA
Issuer DN	C=US;S=UTAH;O=DST;OU=DSTCA;CN=RootCA
Validity Period	05/02/2000 08:00:00 to 05/02/2001 08:00:00
Subject DN	C=US;O=GOV;O=NIH;OU=CIT;CN=Mark Silverman <i>CN=Common name</i>
Subject Public Key	RSA, 3081 8902 8181 ... 0001
Issuer UID	Usually omitted
Subject UID	Usually omitted
Extensions	Optional Extensions
Signature Algorithm	sh1RSA (same as above)
Signature	302C 0258 AE18 7CF2 ... 8D48

Components of the PKI

Come i fanno da IKP? Metti su PKI con
openSSL.

- End Users
- Certification Authorities
- Registration Authorities
- Certificate Directories
- Root CA(s)
- Certification Practice Statements (CPS)
- Certificate Management Protocols & APIs

Major Issues with CAs and RAs

- **End Entity Registration** (Autenticazione dell'end-user: garantire che lui sia chi dichiara di essere)
- **Trust models** ↗ Documento che indica le procedure da seguire che vengono valutate
- **Certification Practice Statement (CPS)**
- **Key management**: generazione di coppie di chiavi. RA non deve conoscere chiave privata
Dove metto chiave privata? Ho dispositivo che mi hardware genera private key
- **Certificate Revocation**: come revoco certificato? Come verifico certificato? Se scade io posso sempre fare controllo sul certificato che però è valido. Devo dire a tutti che certificato non è valido.
- **Publishing Issues**: come ho metto a disposizione
- **Ownership and Maintenance**
- **Liability**: da chi sono le responsabilità

Public-Key Cryptography Standards

- PKCS #1: RSA Cryptography Standard
- PKCS #3: Diffie-Hellman Key Agreement Standard
- PKCS #5: Password-Based Cryptography Standard
- PKCS #6: Extended-Certificate Syntax Standard
- PKCS #7: Cryptographic Message Syntax Standard
- PKCS #8: Private-Key Information Syntax Standard
- PKCS #9: Selected Attribute Types
- PKCS #10: Certification Request Syntax Standard
- PKCS #11: Cryptographic Token Interface Standard
- PKCS #12: Personal Information Exchange Syntax Standard
- PKCS #13: Elliptic Curve Cryptography Standard
- PKCS #15: Cryptographic Token Information Format Standard

Tutte le suole PKCS portano tutti gli standard che appena

• Se ho password usata devo vedere come usarla o per fare il check del certificato

Major Questions

- What mechanisms do users have to trust each other?
- How can users protect the uniqueness of their private key?
- What components of the PKI can be outsourced?
- Who is liable when problems occur?
- How can multiple applications work with each other?

Come si usa una PKI (I)

- **Registrazione di un Utente:**
 - Presso RA l'utente viene identificato
 - Viene prodotta una coppia di chiavi.
 - (generata ad hoc, generata su smartcard, consegnata dall'utente, ...)
 - Viene prodotto il certificato che contiene
 - Informazioni sull'utente
 - Informazioni di certificazione
 - Chiave pubblica
 - Il Certificato viene FIRMATO dalla CA
 - a seconda della polcey*

Come si usa una PKI (II)

- **Firma del Certificato**
 - Firmato dalla CA
- La CA descrive se stessa con un certificato
 - Firmato da se stessa (self-signed)
 - Firmato da un'altra CA

→ La CA ha il suo certificato con chiave pubblica che viene messa a disposizione.

Chi firma il certificato della CA stessa? Ricorda, firmo a self signed, ovvero con la chiave privata che è associata alla chiave pubblica dentro.

I certificati self signed apprezzati dallo stato vanno bene.

Possibile attacco: insomma che chi accetta questo certificato accetta il mio postino italiano.

Come si usa una PKI (III)

- Autenticazione di un Utente
- Descrizione Problema:
 - A vuole autenticare B
 - A si fida di CA
 - ↑ lo ho sul mio sistema
- A conserva il certificato di CA
 - B manda messaggio ad A firmato (firma e certificato) (*ha certificato firmato da CA*)
 - A verifica il certificato (Certificato valido per la sua CA)
 - ↳ Verifica se certificato è valido e non revocato e A verifica la firma della authority
 - A verifica la firma.

B una volta dato il certificato da B, non ha bisogno di richiederlo perché abbiamo già info.

Certificate Revocation

- What constitutes revocation?
- Push/Pull model of CRLs
- Publishing Issues
- Real-time verification?
- Are CRLs the right model?

Revocation Models

Pubblica lista dei certificati revocati quotidianamente.
Revoca alle 07:00 e mette 24 ore a letto
24 ore dopo.

- Certificate Revocation Lists (CRLs)
 - Traditional model
 - Supported by Entrust, Verisign, most CAs
- On-line Certificate Status Protocol (OCSP)
- CRL Distribution Points (CDPs)

Distribuzione delle revoca

New Revocation Models

- On-line Certificate Status Protocol (OCSP)
 - IETF proposed protocol - introduced by VeriSign
 - real-time verification of certificates
 - OCSP responders - provide info to clients
 - acceptance suspended pending response
- Certificate Revocation Trees (Valicert)
 - Offers service and product for real-time verification
 - CRL “trees” - contained within product or at server

Certificate Directories

↑ *acesso e scorsa certificati*

- **Lightweight Directory Access Protocol (LDAP)**
 - runs on TCP/IP, new life into X.500
- Gaining heavy industry support
 - Novell NDS
 - Microsoft AD, Netscape Directory Servers
- Also included in client products
 - msIE, Netscape Communicator
 - etc.

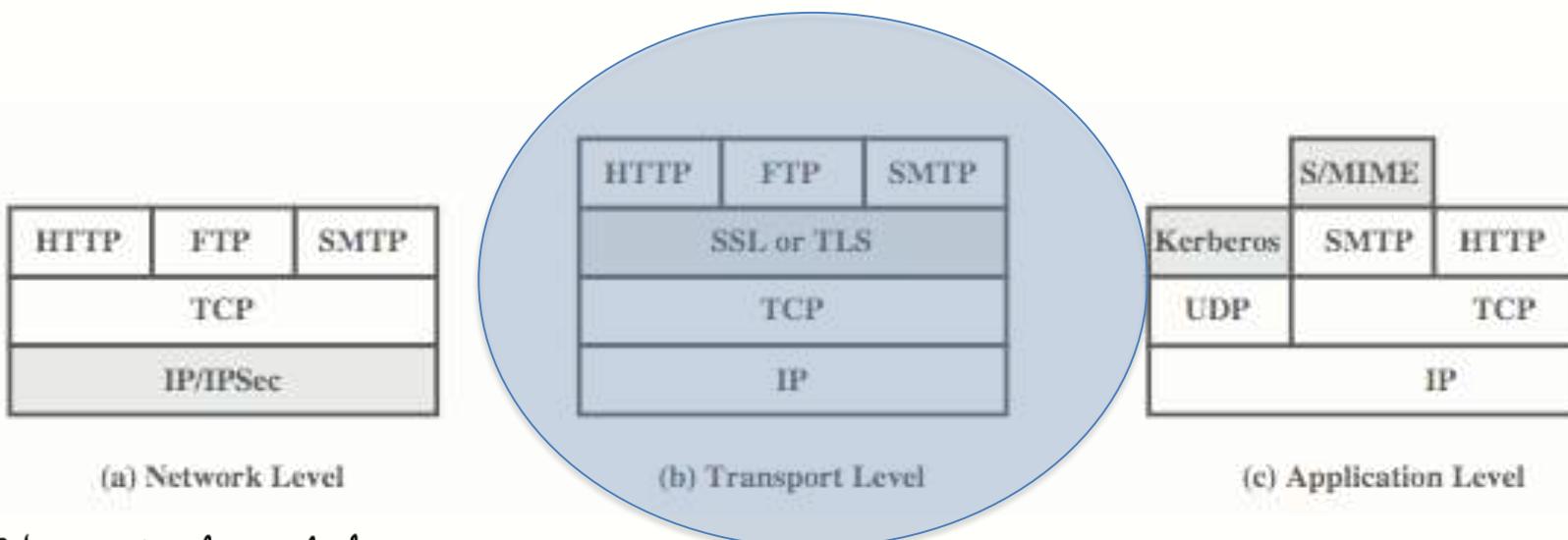
**SICUREZZA A LIVELLO DI
TRASPORTO: **SSL/TSL****

Network Security

Network Security implies securing the communication channels

Tutti i protocolli visti ora sono a livello sopra-applicativo. Posso risolvere a livello rete (IP-Sec) oppure livello trasporti.

Security can affect the channel at one or more layers



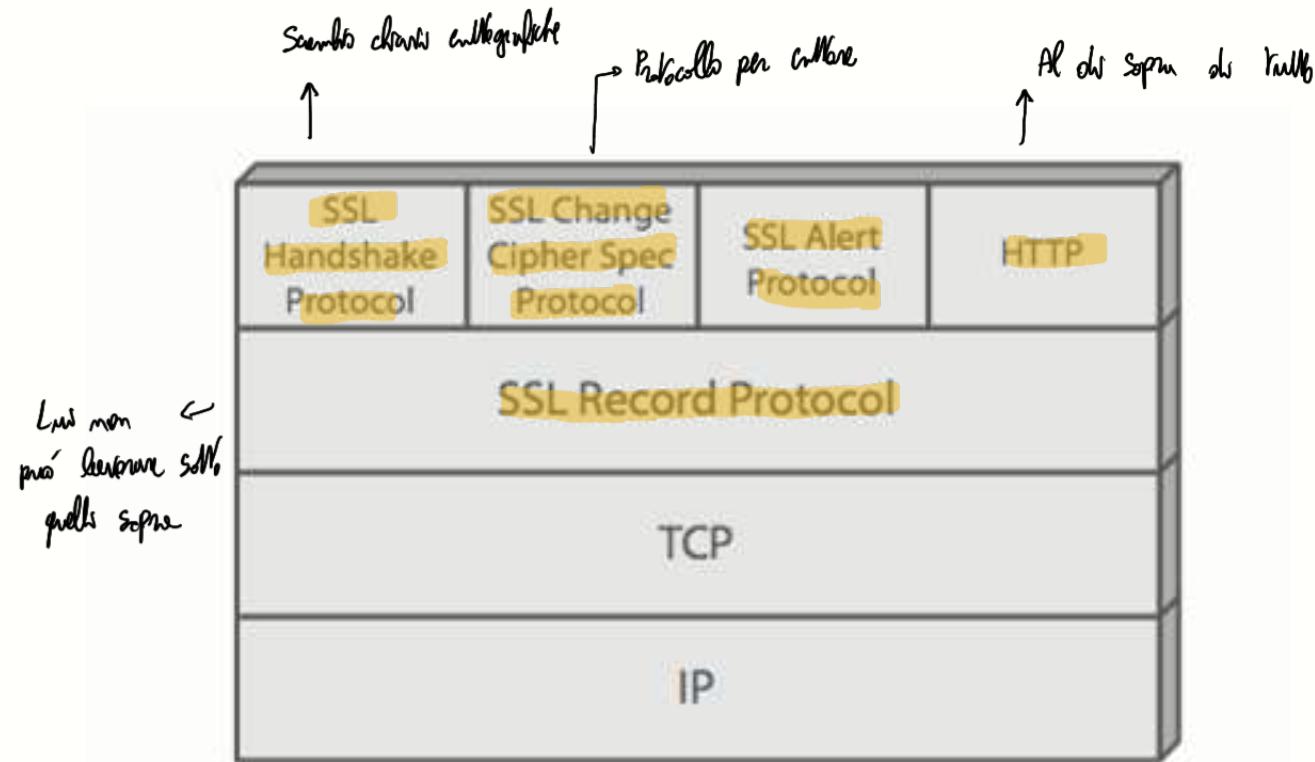
VPN crea tunnel a livello rete, comunicazione dm modo crittato

SSL dm realtivé a livello applicativo, ma per livello applicativo si comporta più come trasporti.

SSL (Secure Socket Layer)

- Transport layer security service
- Originally developed by Netscape
- Version 3 designed with public input
- Subsequently became Internet standard known as TLS (Transport Layer Security)
- Uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

SSL Architecture



SSL Architecture (Pseudo-Transport)

- **SSL connection**
 - a transient, peer-to-peer, communications link
 - associated with 1 SSL session
- **SSL session** (concetto di connessione SSL)
 - an association between client & server
 - created by the Handshake Protocol
 - define a set of cryptographic parameters
 - may be shared by multiple SSL connections

1. Connessione: Relazione temporanea P2P tra 2 parti che fa parte di una sessione SSL.

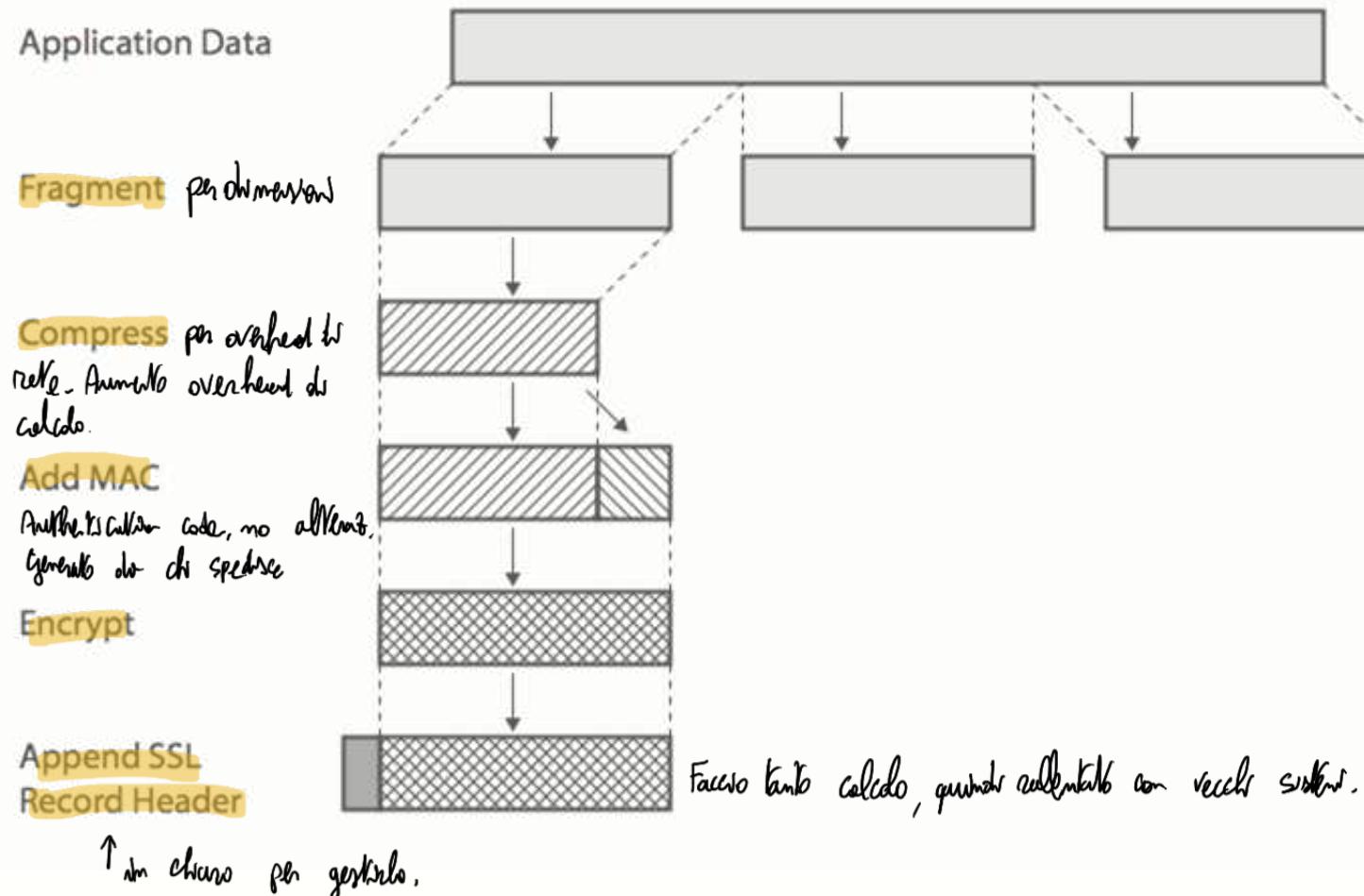
2. Sessione: 2 parti nell'ambito trasporto che scambiano dati

↳ Connessione si alto livello per la quale fanno handshake, costituisce set di parametri crittografici gestiti tra le connessioni, creata sessione per posso aprire tutte le TCP di interesse. Tutti i handshake sono chiusi.

SSL Record Protocol Services

- confidentiality
 - using symmetric encryption with a shared secret key defined by Handshake Protocol
 - AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
 - message is compressed before encryption
- message integrity
 - using a MAC with shared secret key
 - similar to HMAC but with different padding
 - ↳ generally can choose secrets & pos. use hash.

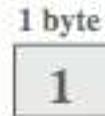
SSL Record Protocol Operation



SSL Change Cipher Spec Protocol

- One of 3 SSL specific protocols which use the SSL Record protocol
- a single message
- causes pending state to become current
- hence updating the cipher suite in use

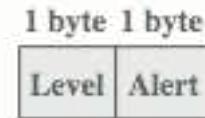
1 message di 1: da adesso in poi userà la suite crittografica.



(a) Change Cipher Spec Protocol

SSL Alert Protocol: *segnala problemi.*

- conveys SSL-related alerts to peer entity
- **severity** *messaggio flaw ordine, qualcosa sbagliata, mac scorretta o errore di decompressione*
 - warning or fatal
- **specific alert**
 - fatal: unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
 - warning: close notify, no certificate, bad certificate, *malfattato*, unsupported certificate, certificate revoked, certificate expired, certificate unknown *(non chiudono al sistema)*
- compressed & encrypted like all SSL data



(b) Alert Protocol

Sarà un numero perché se sono sviluppatori per una sessione privata devi poterli fare.

SSL Handshake Protocol

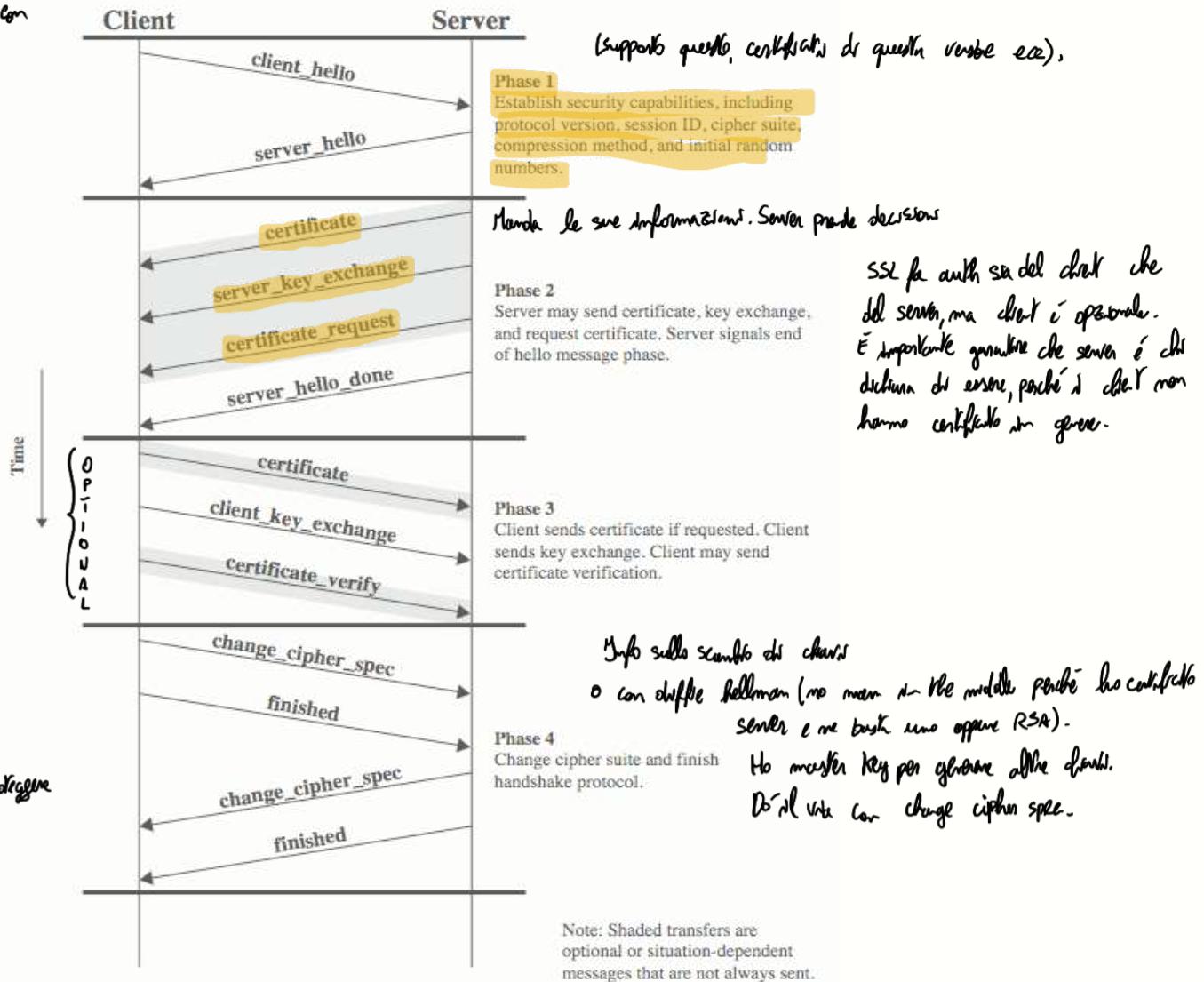
- Allows server & client to:
 - authenticate each other
 - to negotiate encryption & MAC algorithms
 - to negotiate cryptographic keys to be used
- comprises a series of messages in phases
 - Establish Security Capabilities
 - Server Authentication and Key Exchange
 - Client Authentication and Key Exchange
 - Finish

Type	Length	Content
1 byte	3 bytes	≥ 0 bytes

(c) Handshake Protocol

SSL Handshake Protocol

(Per chiedere e poi rifare hello con le scelte più delle supportate)



1 certificato basta a metà: posso proteggere una sessione non busch.

Cryptographic Computations

- Master secret creation
 - a one-time 48-byte value
 - generated using secure key exchange (RSA / Diffie-Hellman) and then hashing info
- Generation of cryptographic parameters
 - client write MAC secret, a server write MAC secret, a client write key, a server write key, a client write IV, and a server write IV
 - generated by hashing master secret

TLS (Transport Layer Security) *combin generație și mac*

- IETF standard RFC 2246 similar to SSLv3
- **with minor differences**
 - in record format version number
 - **uses HMAC for MAC** *de IETF. Restarea incompatibil.*
 - a pseudo-random function expands secrets
 - based on HMAC using SHA-1 or MD5
 - has additional alert codes
 - some changes in supported ciphers
 - changes in certificate types & negotiations
 - changes in crypto computations & padding

HTTPS

- **HTTPS (HTTP over SSL)**
 - combination of HTTP & SSL/TLS to secure communications between browser & server
 - documented in RFC2818
 - no fundamental change using either SSL or TLS
- use **https:// URL rather than http://**
 - and port 443 rather than 80
↓ also use TLS off by default
- **encrypts**
 - URL, document contents, form data, cookies, **HTTP headers**

HTTPS Use

- Connection initiation
 - TLS handshake then HTTP request(s)
- Connection closure
 - have “Connection: close” in HTTP record
 - TLS level exchange close_notify alerts
 - can then close TCP connection
 - must handle TCP close before alert exchange sent or completed

Analisi della scurezza di un server SSL

- Best Practices per la configurazione
- Test di sicurezza



<https://www.ssllabs.com/ssltest/index.html>

SSL Test

The screenshot shows a web browser window with the URL <https://www.ssllabs.com/ssltest/index.html> in the address bar. The page title is "SSL Server Test". A message states: "This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will." Below this is a form with a "Hostname:" input field and a "Submit" button. There is also a checkbox for "Do not show the results on the boards".

Recently Seen

- [jarren.cool](#)
- [pbs-ak.twimg.com](#)
- [up.poznan.pl](#)
- [subscribe.woodworkersjournal...](#)
- [woodflaironfireleu.com.au](#)
- [sogo2.kandou.com](#)
- [webservices.hawaii.sabre.com](#)
- [skylark.up.poznan.pl](#)
- [bact.petermac.org](#)
- [dr-gogle.com](#)

Recent Best

- [devblog.hu](#) A+
- [adc3.transpower.co.nz](#) A
- [mam.iu](#) A
- [www.thedailybeast.com](#) A
- [soap.edue.it](#) A
- [ryley.asoshared.com](#) A
- [gecko.co.uk](#) A
- [distinfit.co](#) A-
- [secureextranet.ethias.be](#) A-
- [mobilecha.themorsegroup.com](#) B

Recent Worst

- [365.wisecrm.com](#) F
- [www.excedorma.com](#) F
- [www.hukl.me](#) T
- [demo-masters.com](#) T
- [www.ta@vd.ch](#) F
- [remote.latimertrnd.co.uk](#) F
- [gtm.customsinfo.com](#) F
- [myaccount.myrepublic.com.sg](#) F
- [crew.lirr.org](#) F
- [st.collegemy.gc.ca](#) F