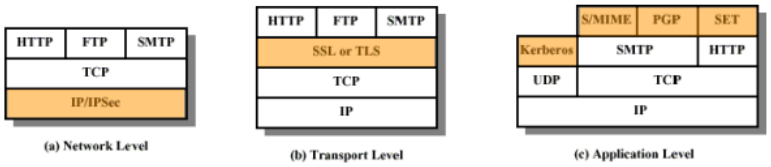
# 09 - Secure Sockets Layer

I protocolli interni sono livellati
Ogni livello provvede servizi ai livelli soprastanti, nascondendo i dettagli dei livelli sotto

- separazione logica
- semplice da sviluppare e mantenere
- interoperabilità
  Due modelli referenziati: ISO/OSI e TCP/IP

- Security in the Internet:
  - at which level?




Sicurezza al livello applicazione:

- pros: definito per le richieste di una specifica applicazione
- cons: richiede molteplici meccanismi di sicurezza

Sicurezza al livello trasporto

- pros: provvede interfacce comuni ai servizi di sicurezza
- cons: richiede (minori) modifiche alle applicazioni

Sicurezza al livello network

- pros: funziona con applicazioni security-ignorant
- cons: può richiedere modifiche al livello OS

## Introduction

Sicurezza al livello applicazione:

- S/MIME
- PGP
- Kerberos
- SET - Secure Electronic Transfer

Sicurezza al livello trasporto:

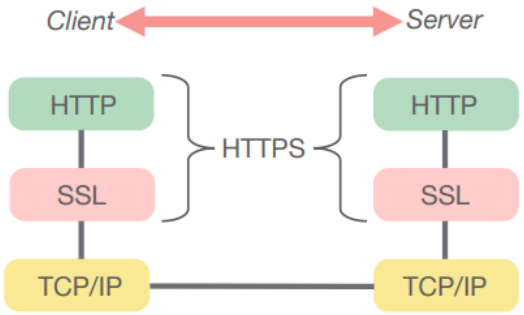
- SSL

Sicurezza al livello network

- IPSec

## SSL

Il servizio di sicurezza più usato su internet
Un servizio generalizzato implementato come set di protocolli che si basano su TCP




Basato su:

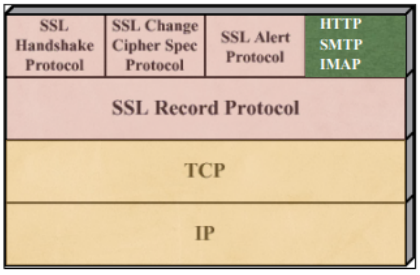
- Cifratura simmetrica
- Cifratura asimmetrica
- Certificati
- Message Authentication Code (MAC)

Soluzione ibrida per la gestione di chiavi

1. $A$ genera $(K_A[\text{pub}], K_A[\text{priv}])$
2. $A$ annuncia la sua chiave pubblica a $B$: $\{K_A[\text{pub}], A\}$
3. $B$ genera la session key $K_S$
4. $B$ invia la session key ad $A$: $C(K_A[\text{pub}], K_S)$
5. $A$ la decripta per ottenere $K_S = D(K_A[\text{priv}], C(K_A[\text{pub}], K_S))$
6. $A$ può cancellare $(K_A[\text{pub}], K_A[\text{priv}])$
7. $A$ e $B$ passano a crittografia simmetrica usando la chiave di sessione $K_S$

## SSL: implementazione

SSL **handshake protocol** crea un canale sicuro, affidabile e autenticato tra client e server.
SSL **record protocol** trasporta messaggi in blocchi incapsulati criptati e autenticati




## SSL: Handshake and Record

- **Handshake**: usa crittografia a chiave pubblica per stabilire un canale sicuro tra i client tale che:
  - ci sia mutua autenticazione
  - client e server concordano sugli algoritmi di criptazione e decriptazione
  - client e server concordano su una chiave segreta
- **Record**: usa crittografia a chiave privata con gli algoritmi e la chiave segreta concordati sopra per scambiare dati confidenzialmente

## SSL: session and connections

SSL sessions:

- una associazione di lunga durata tra client e server
- creato dal protocollo di handshake
- associato a una serie di parametri sicuri
- usato per evitare le costose negoziazioni di nuovi parametri di sicurezza

SSL connections:

- Una connessione trasporto tra client e server
- Le connessioni sono transitorie

- Ogni connessione è associata con una sola sessione

Tra ogni coppia di parti:

- ci possono essere più connessioni
- normalmente una

Session state:

- Session identifier: sequenza di byte arbitraria per identificare una sessione attiva
- Peer certificate: un X.509.v3 certificato del peer (può essere nullo)
- Compression method: metodo usato per comprimere i dati prima della criptazione
- Cipher spec: specifica l'algoritmo di criptazione
- Master Secret: 48 byte secret condiviso tra client e server

Connection state:

- Client/server random: sequenza casuale di byte usata come identificatore scelto dal client e dal server ogni connessione
- Client/server write MAC secret key: Secret key usata nelle operazioni del Message Authentication Code (MAC) sui dati inviati dal client/server
- Client/server write secret key: chiave di criptazione per i dati criptati dal client/server e decriptati dal client/server
- Sequence number

## SSL authentication

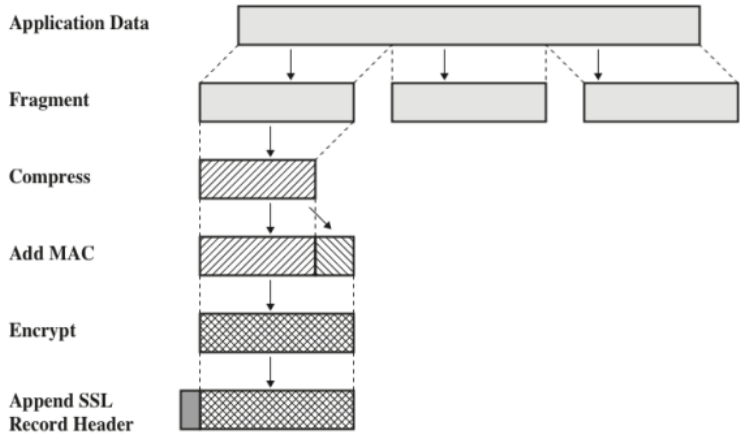
Autenticazione del server dal client tramite certificati è obbligatoria.
Autenticazione del client dal server è opzionale.
Se richiesto dal server, il client normalmente autentica se stesso tramite un meccanismo che non richiede certificati (es login/password), in quanto un certificato SSL è molto costoso (€)

## SSL Record Protocol

I provider di SSL records devono garantire:

- Confidenzialità: L'Handshake Protocol definisce una secret key comune che viene usata per criptare i payload SSL
- Integrità: l'Handshake Protocol definisce una secret key che viene usata per generare Message Authentication Code allegati ai payloads

Il messaggio originale è diviso in $2^{14}$ blocchi di byte.
ogni frammento è numerato, (opzionalmente) compresso, esteso con il MAC, criptato con la master secret key e trasmesso usando TCP
Il ricevente esegue le operazioni in ordine inverso e ricostruisce il messaggio originale che viene passato all'apllicazione al livello superiore (HTTP, SMTP, IMAP, ...)




## MAC in SSL

Ogni frammento è numerato ed esteso da un MAC
MAC è computato come hash (MD5 o SHA-1) del blocco (fragment | seq_no | master secret | padding)
Dove seq_no è lungo 64 bit e quindi non è possibile che si ripeta in una singola sessione

I numeri di sessione rendono inefficaci i replay attacks, inoltre vengono usati per rilevare i blocchi persi (che poi devono essere rigenerati e rispediti)
I MAC sono criptati assieme ai dati usando la crittografia simmetrica con la master secret.

### Importanza dei random bytes

Il "*client hello*", "*server hello*" e pre-master secret messages dell'handshake contengono sequenze di byte randomiche
La segretezza della session key, e quindi della sicurezza del canale di comunicazione creato da SSL dipende tanto sulla randomiticità di questi byte.

La sicurezza dei canali SSL usati dai browser non possono essere più grandi del protocollo più debole della suite di cipher del browser.