



Il protocollo SSL



Il protocollo SSL (*Secure Socket Layer*)

Uno dei protocolli più diffusi nelle comunicazioni sicure:

- garantisce **confidenzialità e affidabilità** delle comunicazioni su Internet, proteggendole da intrusioni, modifiche o falsificazioni.



Il protocollo SSL

- Sviluppato da Netscape per effettuare comunicazioni sicure con il protocollo HTTP
- Prima versione rilasciata nel 1994
- Progettato in modo da permettere la comunicazione tra computer che non conoscono le reciproche funzionalità



SSL

- Utente U che desidera accedere via Internet a un servizio offerto dal sistema S.
- SSL garantisce la **confidenzialità**:
 - la trasmissione è cifrata mediante un sistema ibrido
 - cifrario asimmetrico per costruire e scambiare le chiavi di sessione,
 - cifrario simmetrico che utilizza queste chiavi per criptare dati nelle comunicazioni successive.



SSL

- Il protocollo SSL garantisce l' **autenticazione dei messaggi** accertando l' identità dei due partner,
 - attraverso un cifrario asimmetrico
 - o facendo uso di certificati digitali e inserendo nei messaggi stessi un apposito MAC (che usa una funzione hash one way crittograficamente sicura)



SSL

- SSL si innesta tra un protocollo di trasporto affidabile (TCP/IP) e un protocollo di applicazione (e.g., HTTP).
- SSL è completamente indipendente dal protocollo di applicazione sovrastante.
- Protocollo **HTTPS:**
 - combinazione tra SSL e HTTP
 - utilizzato da Web-server sicuri (prefisso https://...)



SSL

- SSL è organizzato su due livelli:
 1. *SSL Record*
 2. *SSL Handshake*.
- *SSL Record*
 - è al livello più basso
 - connesso direttamente al protocollo di trasporto
 - ha l'obiettivo di incapsulare i dati spediti dai protocolli dei livelli superiori, assicurando **confidenzialità** e **integrità** della comunicazione



SSL

- *SSL Handshake*
 - Responsabile dell'instaurazione e del mantenimento dei parametri usati da *SSL Record*.
 - Permette all'utente e al sistema di
 - autenticarsi,
 - negoziare gli algoritmi di cifratura e firma
 - stabilire le chiavi per i singoli algoritmi crittografici e per il MAC.



SSL

SSL Handshake (meccanismo crittografico)

crea un canale sicuro, affidabile e autenticato tra utente e sistema, entro il quale *SSL Record* fa viaggiare i messaggi divisi in blocchi opportunamente cifrati e autenticati.

SSL Record

realizza *fisicamente* questo canale (meccanismo di rete):

utilizza la cipher suite stabilita da SSL Handshake per cifrare e autenticare i blocchi di dati, prima di spedirli attraverso il protocollo di trasporto sottostante



SSL Handshake

- Una sessione di comunicazione è innescata da uno scambio di messaggi preliminari (*handshake*) indispensabili per la creazione del canale sicuro
- attraverso questi messaggi il sistema S (*server*) e l'utente U (*client*)
 - si identificano a vicenda
 - cooperano alla costruzione delle chiavi segrete da impiegare nelle comunicazioni simmetriche successive.
- Il protocollo è organizzato in passi.



1. Utente: *client hello*

U manda a S un messaggio di *client hello*, con cui

- richiede la creazione di una connessione SSL,
- specifica le prestazioni di “sicurezza” che desidera siano garantite durante la connessione
 - cifrari e meccanismi di autenticazione che U può supportare
- e invia una sequenza di byte casuali.



ESEMPIO: *cipher suite*

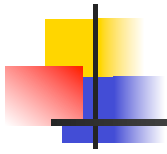
SSL_RSA_WITH_AES_CBC_SHA1

RSA per scambio chiavi di sessione

AES per cifratura simmetrica

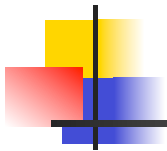
CBC indica l'impiego di un cifrario a composizione di blocchi

SHA1 funzione hash one-way per il MAC



2. Sistema: *server hello*

- Il sistema S
 - riceve il messaggio di *client hello*
 - seleziona una *cipher suite* che anch'esso è in grado di supportare
 - invia a U un messaggio di *server hello* che specifica la sua scelta e contiene una nuova sequenza di byte casuali.
- Se U non riceve il messaggio di *server hello* interrompe la comunicazione.



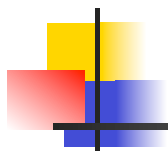
3. Sistema: invio del certificato

- S si autentica con U inviandogli il proprio certificato digitale
 - e gli eventuali altri certificati della catena di certificazione dalla sua CA fino alla CA radice
- Se i servizi offerti da S devono essere protetti negli accessi, S può richiedere a U di autenticarsi inviando il suo certificato digitale.
 - avviene raramente: la maggior parte degli utenti non ha un certificato personale,
 - il sistema dovrà accertarsi dell'identità dell'utente in un secondo tempo.



4. Sistema: *server hello done*


S invia il messaggio *server hello done* con cui sancisce la fine degli accordi sulla *cipher suite* e sui parametri crittografici a essa associati.



5. Utente: autenticazione del sistema

Per accertare l'autenticità del certificato ricevuto da S, l'utente U

- controlla che la data corrente sia inclusa nel periodo di validità del certificato
- che la CA che ha firmato il certificato sia tra quelle "fidate"
- e che la firma apposta dalla CA sia autentica



6. Utente: invio del *pre-master secret* e costruzione del *master secret*

L'utente U


- costruisce un *pre-master secret* costituito da una nuova sequenza di byte casuali (i.e., genera un numero segreto)
- lo cifra con il cifrario a chiave pubblica su cui si è accordato con S
- spedisce il relativo crittogramma a S

(e.g., U cifra il premaster secret con RSA, usando la chiave pubblica presente nel certificato di S)




6. Utente: invio del *pre-master secret* e costruzione del *master secret*

- Il *pre-master secret* viene combinato da U con alcune stringhe note e con i byte casuali presenti sia nel messaggio di *client hello* che in quello di *server hello*.
- A tutte queste sequenze U applica delle funzioni hash one-way (SHA-1 e MD5) secondo una combinazione opportuna.
- Ottiene così il *master secret*



7. Sistema: ricezione del pre-master secret e costruzione del master secret

- S decifra il crittogramma contenente il *pre-master secret* ricevuto da U
- S calcola il *master secret* mediante le stesse operazioni eseguite da U al passo 6 (dispone delle stesse informazioni).



7. Sistema: ricezione del pre-master secret e costruzione del master secret

- Sia U che S conoscono:
 - Il numero casuale che U ha mandato a S (inviato in chiaro)
 - Il numero casuale che S ha mandato a U (inviato in chiaro)
 - Il *premaster secret* (inviato cifrato)
- U e S calcolano il **master secret**.



8. Utente: invio del certificato (opzionale)

- Se all'utente U viene richiesto un certificato (passo 3) ed egli non lo possiede il sistema interrompe l'esecuzione del protocollo.
- Altrimenti U invia il proprio certificato con allegate una serie di informazioni firmate con la sua chiave privata, tra cui
 - il *master secret*
 - tutti i messaggi scambiati fino a quel momento (*SSL-history*)



8. Utente: invio del certificato (opzionale)

- S controlla il certificato di U e verifica autenticità e correttezza della *SSL-history*.
- In presenza di anomalie, la comunicazione con U viene interrotta.



9. Utente/Sistema: messaggio *finished*

- È il primo messaggio protetto mediante il *master secret* e la *cipher suite* su cui i due partner si sono accordati.
- Il messaggio viene prima costruito da U e spedito a S, poi costruito da S e spedito a U:
 - nei due invii la struttura del messaggio è la stessa, ma cambiano le informazioni in esso contenute.



9. Utente/Sistema: messaggio *finished*

- La costruzione avviene in due passi:
 - all'inizio si concatenano il *master secret*, tutti i messaggi di *handshake* scambiati fino a quel momento e l'identità del mittente (U o S)
 - la stringa ottenuta viene trasformata applicando le funzioni SHA-1 e MD5: si ottiene una coppia di valori che costituisce il messaggio *finished*.



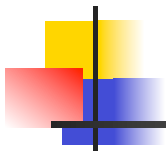
9. Utente/Sistema: messaggio *finished*

- Il messaggio è diverso nelle due comunicazioni perché S aggiunge ai messaggi di *handshake* anche il messaggio *finished* ricevuto da U.
- Il destinatario della coppia (S o U) non può invertire la computazione precedente in quanto generata da funzioni *one-way*:
 - ricostruisce l'ingresso delle due funzioni SHA-1 e MD5, le ricalcola e controlla che la coppia generata coincida con quella ricevuta, come dimostrazione che la comunicazione è avvenuta correttamente.



SSL *handshake*

- Il master secret è utilizzato da U e da S per costruire una propria tripla contenente
 - la chiave segreta da adottare nel cifrario simmetrico
 - la chiave per l'autenticazione del messaggio (costruzione del MAC)
 - la sequenza di inizializzazione per cifrare in modo aperiodico messaggi molto lunghi (usata e.g., come valore iniziale nel CBC).
- Le triple di U e di S sono diverse tra loro ma note a entrambi i partner: ciascuno usa la propria, il che aumenta la sicurezza delle comunicazioni.



SSL

Il canale sicuro approntato dal protocollo *SSL handshake* viene realizzato dal protocollo *SSL record*.

- I dati sono *frammentati in blocchi*.
- **Ciascun blocco viene**
 - numerato, compresso e autenticato mediante l'aggiunta di un MAC
 - cifrato mediante il cifrario simmetrico su cui U e S si sono accordati
 - trasmesso dall'*SSL record* utilizzando il protocollo di trasporto sottostante.



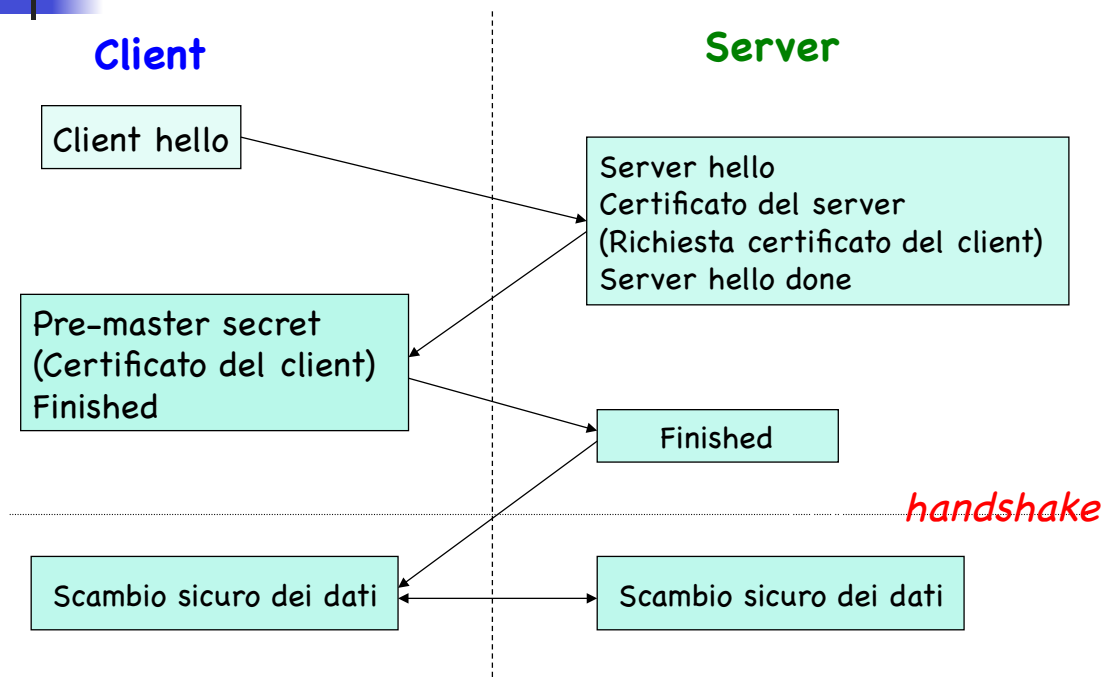
SSL

Il destinatario esegue un procedimento inverso sui blocchi ricevuti:

- *decifra e verifica la loro integrità attraverso il MAC*
- **decomprime e riassembla i blocchi in chiaro**
- li consegna all'applicazione sovrastante.



Protocollo SSL



Sicurezza: *Client hello e server hello*

Nei passi di *hello* i due partner creano e si inviano due sequenze casuali per la costruzione del *master secret*, che risulta così diverso in ogni sessione di SSL.

Un crittoanalista non può riutilizzare i messaggi di *handshake* catturati sul canale per sostituirsi a S in una successiva comunicazione con U.



MAC dei blocchi di dati

- SSL *record* numera in modo incrementale ogni blocco di dati e autentica il blocco attraverso un MAC.
- Per prevenire la modifica del blocco da parte di un crittoanalista attivo, il MAC viene calcolato come immagine hash one-way di una stringa costruita concatenando
 - il contenuto del blocco,
 - il numero del blocco nella sequenza,
 - la chiave del MAC
 - alcune stringhe note e fissate a priori.



MAC dei blocchi di dati

Dato che i MAC sono cifrati insieme al messaggio, un crittoanalista non può alterarli senza avere forzato prima la chiave simmetrica di cifratura:

un attacco volto a modificare la comunicazione tra i due partner è difficile almeno quanto quello volto alla sua decrittazione.



Il sistema è autenticato

- Il canale definito da SSL *handshake* è immune da attacchi attivi *man-in-the-middle* poiché il sistema S viene autenticato con un certificato digitale.
- L'utente U può comunicare il *pre-master secret* al sistema S in modo sicuro attraverso la chiave pubblica presente nel certificato di S.
- Solo S può decifrare quel crittogramma e costruire il *master secret*, su cui si fonda la costruzione di tutte le chiavi segrete adottate nelle comunicazioni successive.
- Solo il sistema S, quello cui si riferisce il certificato, potrà quindi entrare nella comunicazione con l'utente U.



L'utente può essere autenticato

Il certificato di U (se richiesto) e la sua firma apposta sui messaggi scambiati nel protocollo (*SSL-history*) consentono a S di verificare che U sia effettivamente quello che dichiara di essere e che i messaggi siano stati effettivamente spediti da esso.

Se ciò non si verifica, S deduce che il protocollo è stato alterato (casualmente o maliziosamente con un attacco *man-in-the-middle*) e interrompe la comunicazione.



L'utente può essere autenticato

- L'opzionalità dell'autenticazione dell'utente ha reso l'SSL molto diffuso nelle transazioni commerciali via Internet:
 - per gli utenti la necessità di certificazione può costituire un ostacolo pratico ed economico
- L'utente può essere autenticato con altri metodi (*login e password*, # carta di credito)



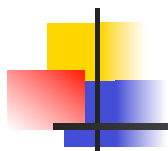
Generazione delle sequenze casuali

- Le tre sequenze casuali generate da U e da S e comunicate nei messaggi di *client hello*, *server hello*, *pre-master secret* sono usate per creare il *master secret*, quindi per generare le chiavi segrete di sessione.
- La sequenza corrispondente al *pre-master secret* viene generata da U e comunicata per via cifrata a S.
- La non predicibilità di questa sequenza è cruciale per la sicurezza del canale SSL:
 - una sua cattiva generazione renderebbe il protocollo molto debole.



Messaggio *finished*

- Contiene tutte le informazioni scambiate nel corso dell' *handshake*
- *Scopo*: consente un ulteriore controllo sulle comunicazioni precedenti per garantire che
 - queste siano avvenute correttamente
 - U e S dispongano dello stesso *Master Secret*
 - che la comunicazione non sia stata oggetto di un attacco attivo



SSL

- È almeno sicuro quanto il più debole *cipher suite* supportato
- dal 2000 le norme internazionali non pongono alcuna limitazione sui cifrari impiegabili (se non in alcuni paesi)
- è consigliabile disabilitare i propri sistemi dall'impiego di cifrari ormai notoriamente insicuri e chiavi troppo corte