Client

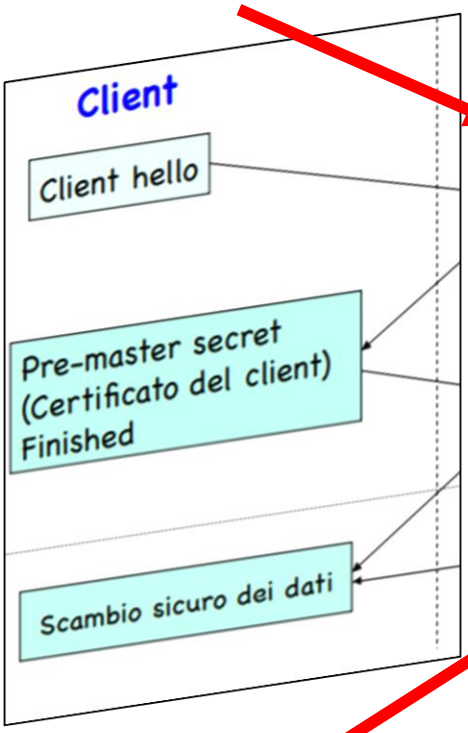
Server

Messaggio *client hello*

L'utente U invia al sistema S un messaggio con il quale:

- si richiede la creazione di una connessione SSL;
- specifica i cifrari e meccanismi che supporta e le prestazioni di sicurezza richieste;
- invia una sequenza di byte casuali.

Tutto questo è mandato in chiaro!



Messaggio *server hello*

Il sistema S riceve il messaggio dell'utente U.

- Seleziona una cipher suite che anche lui supporta (cerca di soddisfare le richieste dell'utente U)
- Invia un messaggio all'utente U dove specifica la sua scelta.
- Appende dei byte casuali alla risposta.

Anche questo è mandato in chiaro!

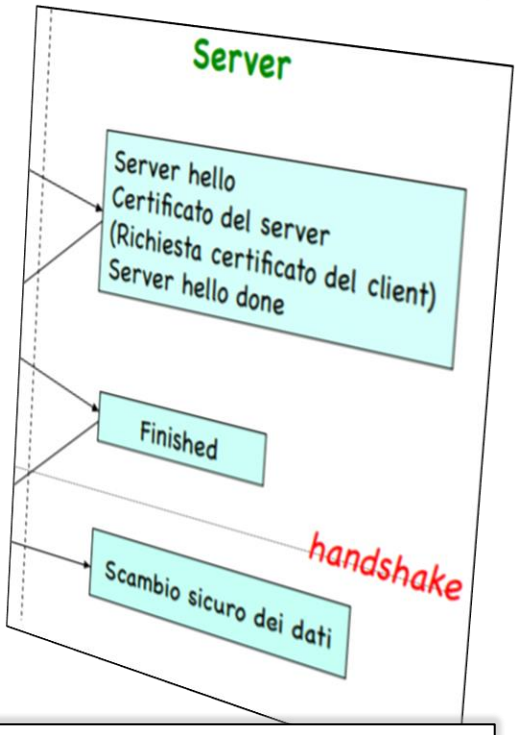
Autenticazione

Il sistema S si autentica con U inviandogli il proprio certificato digitale (e gli eventuali altri certificati fino al primo nodo comune nella catena delle CA).

NB Se i servizi offerti da S devono essere protetti negli accessi anche S può richiedere a U di autenticarsi inviando il suo certificato digitale. Avviene raramente in quanto la maggior parte degli utenti non ha un proprio certificato e in genere ci si accerta dell'identità di un utente in un secondo modo (autenticazione sul sito web ad esempio).

Messaggio *server hello done*

Messaggio con il quale il server S sancisce la fine degli accordi sulla cipher suite ed i parametri crittografici associati.



Controllo da parte del client

L'utente U accerta l'autenticità del certificato ricevuto dal sistema S tramite la data, tramite la CA che lo ha firmato, ecc. Estrae la chiave pubblica dal certificato. L'utente U:

- costruisce il *pre-master secret* costituito da una nuova sequenza casuale di byte;
- lo cifra con la chiave pubblica estratta dal certificato.
- spedisce il crittogramma al sistema S.

Costruzione del *master secret*

Sempre l'utente U costruisce il *master secret* partendo da:

- il *pre-master secret* (cifrato con RSA, si usa la chiave pubblica presente nel certificato di S);
- i byte casuali di *client hello* e *server hello* (*client hello* lo ha di suo, mentre *server hello* è stato inviato dal server).

Applica a tutte queste sequenze delle funzioni hash one-way secondo una combinazione opportuna. Il nuovo valore ottenuto è il *master secret*.

Ricostruzione del *master secret*

Anche il sistema S si calcola localmente il master secret. Può farlo perchè possiede:

- il *pre-master secret*, appena ricevuto dall'utente U e decriptato con la sua chiave privata
- i byte casuali di *client hello* e *server hello* (*client hello* lo ha ricevuto dall'utente U, mentre *server hello* lo ha di suo)

Entrambi gli utenti hanno il *master secret*!

Messaggio *finished*

E' il primo messaggio protetto da *master secret* e *cipher suite* accordati. Il messaggio è costruito dall'utente U e inviato al sistema S, poi costruito dal sistema S ed inviato all'utente U. Il messaggio ha la stessa struttura ma cambiano i dati (la history).