Azzolini Riccardo 2021-05-20

SSL/TLS

1 SSL/TLS

SSL (**Secure Socket Layer**) / **TLS** (**Transport Layer Security**) fornisce servizi di riservatezza, integrità e autenticazione tra due applicazioni che comunicano via Internet, ed è usato principalmente nello scenario client-server, in particolare per proteggere le informazioni trasmesse tra un browser e un server Web.

Il nome TLS fa riferimento al livello di trasporto perché esso protegge appunto le informazioni scambiate al livello di applicazione, ma tecnicamente il servizio di sicurezza SSL/TLS stesso rientra nel livello di applicazione, dato che è implementato sopra il protocollo di trasporto TCP.

SSL fu inizialmente progettato da Netscape, fino alla versione 3 che fu progettata con revisione pubblica. Successivamente esso è diventato uno standard IETF chiamato TLS, di cui sono state definite varie versioni, tra le quali quella attualmente più usata è TLS 1.2, mentre la più recente è TLS 1.3 (pubblicata nel 2018). Le principali differenze tra le versioni fino a TLS 1.2 sono gli algoritmi supportati e la resistenza ad attacchi noti, mentre TLS 1.3 apporta alcune modifiche alla struttura di una parte del protocollo (l'handshake).

2 Servizi di sicurezza forniti

SSL/TLS protegge i dati scambiati tra le applicazioni (ad esempio, il protocollo HTTPS usati per la sicurezza in ambito Web è HTTP implementato sopra TLS, "HTTP over TLS", e TLS si può usare anche per FTP, email, ecc.), ma non protegge contro attacchi eseguiti ai livelli inferiori (ad esempio spoofing e DoS al livello di rete / IP). In particolare, esso realizza:

- l'autenticazione tramite certificati dell'identità del server, ed eventualmente del client;
- la riservatezza (cifratura), autenticazione e integrità di tutti i dati scambiati tra server e client.

3 Sessioni e connessioni

La comunicazione SSL/TLS tra due host è modellata da due relazioni, entrambe temporanee ma di durate diverse:

- una **connessione** SSL/TLS identifica una relazione transiente di trasporto dei dati tra nodi (ad esempio una connessione HTTP);
- una **sessione** SSL/TLS è un'associazione tra client e server creata tramite il protocollo handshake (quello con cui si negoziano i parametri di sicurezza).

Una connessione è associata a una sola sessione, mentre in una sessione si possono avere più connessioni: per tutta la durata della sessione i parametri di sicurezza negoziati nell'handshake rimangono validi e possono essere usati per più connessioni.

4 Protocolli SSL/TLS

SSL/TLS non è un singolo protocollo, ma è costituito da un insieme di quattro protocolli, che sono organizzati in due livelli.

Il protocollo di livello inferiore è **Record**, che stabilisce come strutturare e processare i messaggi per garantire la sicurezza dei protocolli dei livelli superiori (sia gli altri protocolli SSL/TLS che i veri e propri protocolli applicativi come HTTP).

I tre protocolli di livello superiore sono invece:

- **Handshake**, che negozia i parametri di sicurezza e autentica server e client;
- **Change Cipher Spec**, un protocollo costituito da un unico messaggio che sincronizza la chiusura dell'handshake;
- **Alert**, che gestisce gli errori.

Tutti i dati applicativi e i messaggi dei tre protocolli di livello superiore, compresi quelli inviati all'inizio dell'handshake, prima ancora di stabilire i parametri i parametri di sicurezza, sono strutturati secondo il protocollo Record.

5 Protocollo Record

Il protocollo Record implementa servizi di

- *riservatezza*, tramite la cifratura simmetrica;
- *integrità* e *autenticazione*, tramite un MAC.

Per la cifratura e per il MAC si usano due diverse chiavi segrete,[1] entrambe condivise mediante il protocollo Handshake. Inoltre, le chiavi sono legate alla direzione del flusso, dunque una sessione SSL/TLS richiede in totale quattro chiavi segrete condivise.

Per trasmettere dei dati il protocollo Record esegue i seguenti passi:

1. frammenta l'input in blocchi di 2^{14} byte;
2. opzionalmente comprime i dati;
3. computa un codice MAC (ad esempio HMAC) sul blocco;
4. cifra il blocco con un algoritmo simmetrico;
5. aggiunge un'intestazione nella quale sono specificati:
   - il tipo di protocollo di livello superiore (ad esempio Handshake, Alert, HTTP, ecc.);
   - la lunghezza in byte del frammento;
   - la versione di SSL/TLS;
6. trasmette il risultato in un segmento TCP.

Si osservi che l'autenticazione viene applicata sul testo in chiaro, prima della cifratura. Il calcolo del MAC comprende però anche alcune delle informazioni presenti nell'header (nonostante questo venga aggiunto al messaggio solo dopo la cifratura):

- il numero di sequenza del messaggio (`seq_num`);
- il tipo di protocollo di livello superiore (`TLSCompressed.type`);
- la versione di TLS (`TLSCompressed.version`);
- la lunghezza del frammento (`TLSCompressed.length`);
- il frammento vero e proprio, ovvero i dati del protocollo di livello superiore (`TLSCompressed.fragment`).

In simboli, il calcolo del MAC è espresso dalla formula

$$\begin{aligned}\mathrm{MAC}(&\texttt{MAC\_write\_key},\ \texttt{seq\_num} \parallel \texttt{TLSCompressed.type} \parallel \texttt{TLSCompressed.version} \\ &\parallel \texttt{TLSCompressed.length} \parallel \texttt{TLSCompressed.fragment})\end{aligned}$$

dove l'operatore $\parallel$ indica la concatenazione e `MAC_write_key` è la chiave usata da questo host per generare i MAC (la quale è diversa dalla chiave usata per verificare i MAC perché le chiavi sono legate alla direzione del flusso).

[1] Usare chiavi diverse per implementare servizi di sicurezza diversi non è una caratteristica specifica di SSL/TLS, ma è un principio che è bene applicare in generale. Ad esempio, anche in IPSec si usano SA diverse (e quindi chiavi diverse) per fornire i servizi di riservatezza e di autenticazione.

Quando il protocollo Record viene usato per trasportare i messaggi di Handshake, prima che siano stabiliti gli algoritmi crittografici e le chiavi da usare, esso esegue la frammentazione, la compressione (opzionale) e l'aggiunta dell'header, ma non applica il MAC e la cifratura.

6 Protocollo Alert

Il protocollo Alert viene usato per inviare appunto messaggi di alert durante l'esecuzione di TLS. Un messaggio di alert è costituito da due byte:

- il primo byte indica la tipologia di alert, che può essere *warning* (indicato dal valore 1) o *fatal* (valore 2);
- il secondo byte specifica l'errore che questo messaggio di alert segnala.

Gli alert di tipo fatal comportano la chiusura immediata della sessione. Alcuni esempi sono:

- messaggio non conforme al protocollo;
- MAC di un messaggio Record non valido;
- errore nell'handshake.

Invece, due tipici esempi di warning sono "certificato scaduto" e "certificato sconosciuto". I warning sono gestiti dal livello applicativo, che può scegliere di proseguire accettando i relativi rischi oppure di interrompere la sessione, eventualmente lasciando decidere all'utente (come spesso fanno i browser). Ciò consente una maggiore flessibilità, ma significa che pur usando SSL/TLS sono possibili sessioni non sicure.

Siccome i messaggi di alert sono trasportati dal protocollo Record, essi vengono compressi e cifrati come tutti gli altri dati SSL/TLS.

7 Protocollo Change Cipher Spec

Il protocollo Change Cipher Spec prevede un unico messaggio, costituito da un singolo byte contenente il valore 1 (che viene trasportato dal protocollo Record). Il suo scopo è rendere definitiva la scelta degli schemi crittografici negoziati durante l'handshake, che vengono usati per tutti i messaggi successivi a questo.