1 Overview about TLS and IPsec

TLS ha come progenitore il protocollo SSL da un certo momento in poi cambiò nome a seguito di importanti aggiornamenti al protocollo. Nelle varie versioni di TLS hanno aggiunto notevoli funzionalità aggiuntive tra cui il supporto al protocollo di trasporto dati UDP, l'utilizzo (per poi esser abbandonati in TLV v.1.3) di algoritmi di encrypt come MD5 e SHA-1 ed infine, in TLS v.1.3 (la versione ad oggi in uso) di utilizzare i protcolli AEAD:protocolli che garantiscono confidenzialità e integrità dei dati. Dato che TLS utilizza TCP (o nella versione DTLS UDP), si assegna ad ogni processo una socket per rendere sicuri i dati di livello applicativo è considerato un protocollo che opera tra il livello di trasporto e il livello applicativo. Il suo nome, quindi, è fuorviante *Transport Layer Security*. Come vedremo in seguito questa possibilità di aprire una socket per ogni applicazione risulta problematica. Infine, TLS protegge solamente il payload di TCP e non l'intero pacchetto poiché per come è stato progettato, se così non fosse, non si saprebbe dove e a chi inviare i pacchetti. Da questo aspetto se ne conclude che l'header del pacchetto TCP può essere modificato e quindi si è soggetti ad attacchi di tipo TCP spoofing, MITM, CCA etc...
Un altro protocollo, sviluppato di pari passi a TLS, è IPsec. Esso viene considerato una versione molto più crittograficamente sicura e, data la problematica di TLS nell'utilizzare socket diverse per le varie applicazione, si pone tra il livello di trasporto e di rete. Infatti, "monta"sopra IP e, grazie a questa proprietà, è possibile rendere crittograficamente sicuro l'intero pacchetto TCP/UDP/Altro incapsulando il pacchetto crittografato IP in un altro paccheto così da nascondere l'intero contenuto di quello che si voleva mandare.
Da qui il concetto di **traffic flow confidentiality** che rappresenta un nuovo requisito di sicurezza:la straficazione del pacchetto deve aggiungere sicurezza al protocollo.
Una differenza tra TLS e IPsec è che nella loro progettazione hanno sviluppato il set-up dell'ambiente e il modo in cui si trasferiscono i dati in maniera differente. Questi due aspetti, infatti, rappresenzanto due concetti cardini nello sviluppo di un protocollo di sicurezza.
Quindi, TLS decise di unire il set-up dell'ambiente e il trasferimento di dati delegano il primo aspetto venne realizzato tramite una fase di handshake in cui si negoziano gli algoritmi e, tramite crittografia asimmetrica, si comunicano tali chiavi, necessarie a garantire integrità e confidenzialità, la seconda fase, invece viene detta record phase;mentre in IPsec, si decise di disaccoppiare questi due aspetti: la prima fase di set-up è delegata ad un protocollo automatico detto IKE (Internet Key Exchange);la seconda fase, di trasferimento dati delegata al protocollo utilizzato a supporto di IPsec.

2 TLS Protocol Stack

TLS, nella record phase, utilizza un protocollo detto TLS Protocol Stack che è composto da diverse componenti dette:TLS Record Protocol responsabile del trasferimento dei dati, Handshake Protocol responsabile dello scambio delle informazioni necessarie alla sicurezza della comunicazione, Alert Protocol responsabile della definizione dei messaggi di warning/alert, il Change Cipher Suite responsabile dell'inizializzazione del cipher. Il Protocol Stack si interpone tra HTTP e TCP.

3 TLS Record Protocol

Il TLS Record Protocol è responsabile del trasferimento dei dati dell'utente e, in base agli obiettivi proposti da TLS, a questi dati bisogna aggiungere integrità e confidenzialità. In aggiunte, per questioni tecniche, si decise di aggiungere anche una funzione di compressione dei dati. In totale, quindi vi sono queste funzionalità: compressione, HMAC (per l'integrità) e Encryption. Quello che si potrebbe pensare è che non vi è alcuna differenza tra l'ordine in cui vengono effettuate queste operazioni, tuttavia non è così:ne è un esempio l'attacco CRIME in cui si frutta la compressione e poi l'encryption per decifrare l'intero dato che abbiamo ricevuto. Inoltre, un'altra operazione totalmente scorretta è quella di effettuare prima l'integrità con HMAC e poi l'encryption.Alla fine delle tre fasi (compression,integrity,encryption **IN QUESTO ORDINE**) si aggiunge un record header composto da Content Type,Major Version, Minor Version,Compressed Lenght.Questo header è formato da 5 bytes.

4 Message Authentication Code

Il Message Authentication Code rappresenta la parte del pacchetto TLS che serve a fornire integrità.TLS scelse di adottare HMAC per le sue proprietà di difesa contro attacchi di tipo Message Spoofing e MITM. Tuttavia, non si è protetti contro attacchi di replay attack, cioè attacchi in cui si reinvia un messaggio o una porzione di esso per indurre comportamenti non desiderati. Per proteggersi da questo tipo di attacco bisogna aggiungere al protocollo una informazione che aggiunge freshness alla comunicazione. TLS decise di adottare il NONCE. Il NONCE è una serie di bit casuali che cambiano in ogni comunicazione e può essere inviato in chiaro poiché non costituisce un requisito di sicurezza. In TLS per ottenere un NONCE che cambia in ogni comunicazione, scelsero di utilizzare un numero di sequenza scambiato tra Client e Server. Inoltre, per motivi di ottimizzazione e ricordando che il protocollo TCP utilizza già dei propri numeri di sequenza per fornire un servizio affidabile di trasporto dati, decisero di legare TLS e TCP utilizzando gli stessi numeri di sequenza. Da qui nasce l'esigenza di sviluppare DTLS (Datagram Transport Layer Security) per supportare UDP e la sua natura di trasporto non affidabile di datagrammi.

In conclusione, quindi le operazioni che vengono effettuate per aggiungere integrità al pacchetto è quella di prendere i dati, affiancare un TLS Header e prima di esso il sequence number che, visto che viene trasmesso direttamente in TCP, non viene trasmesso.

5 Encyption and Authentication

L'utilizzo dell'encryption permette di assicurare la sicurezza semantica, mentre l'authentication permette di avere integrità ed,in particolare, si ottiene la cosiddetta proprietà di unforgeability che consente di rendere inforgiabili i messaggi (protezione da message tampering e message spoofing) se non si possiede la chiave. L'unica encryption che consente di ottenere sia sicurezza semantica che integrità sono gli algoritmi AEAD (i.e.Ephimeral Diffie-Hellman) . Per garantire questi aspetti in un protocollo di sicurezza bisogna scegliere fa cifrare e poi autenticare o viceversa. In particolare, si ha che in SSL (la versione peggiore) si è scelto di cifrare il dato orginario e poi effettuare l'autenticazione proprio sul dato originario. La soluzione che garantisce un livello di sicurezza maggiore, ma non comunque corretta, è quella proposta in TLS in cui si effettua l'integrità sul dato cifrato. Infine, la soluzione che fino ad ora è risultata la migliore è quella proposta da IPsec in cui prima si cifra il dato da inviare per poi aggiungere l'integrità sul ciphertext.

6 Block Ciphers:Introduzione

I Block cipher rappresentano un'alternativa ai stream cipher. In particolare, come dice il nome, lavorano in blocchi: preso un plaintext lo si divide in blocchi a cui verrà applicata una certa funzione PRF (Pseudo Random Function) per ottenere un ciphertext; in seguito, per decifrare il ciphertext, si effettua l'operazione inversa applicando PRF^{-1} per ottenere tutti i blocchi di plaintext che, ricongiunti, formeranno il messaggio inizialmente inviato. Un PRF è una pseudo random function che consiste nel considerare tutte le permutazioni di un insieme e si sceglie una permutazione di essa in maniera uniforme. Il vincolo di queste PRF è che la funzione deve essere biettiva e quindi due elementi del dominio devono avere due elementi diversi nel codominio. Nel mondo reale, questa probabilità deve essere negligible. Inoltre, nel caso in cui il messaggio è superiore di un blocco, si divide il messaggio e si cifrano indipendentemente i singoli blocchi. Questo tipo di algoritmo viene detto ECB (Electronic Code Book) e non deve mai essere utilizzato per le seguenti ragioni:preso un plaintext, porzioni di messaggio, sono codificate nello stesso ciphertext; preso un plaintext, se esso viene cifrato due volte si produrrà lo stesso plaintext. Per risolvere queste problematiche si deve quindi introdurre una freshness data da IV (bit casuali lunghi quanto il blocco) che, nel momento dell'encryption viene XORato con il plaintext, e poi introdotto all'interno del PRF;nel momento della decyption, il ciphertext viene inserito all'interno del PRF^{-1} per poi essere XORato con l'IV.

Condizione necessaria affinché è possibile decriptare è che si invii l'IV insieme al ciphertext.
La maniera più corretta di usare un block cipher è quella di prendere il messaggio, dividerlo in blocchi e a ciascun blocco inserire un IV che viene XORato con il plaintext, e poi introdotto nel PRF per generare il ciphertext. Ad ogni blocco del ciphertext si affianca il relativo IV, viene compattato ed inviato a destinazione. Nella decryption si effettua il procedimento inverso.
In conclusione, se dobbiamo necessariamente utilizzare ECB dobbiamo usarlo per messaggi molto piccoli e che non si ripetano. In tutti gli altri casi, si utilizzano i modes of operation: sono costruzioni particolari necessarie per utilizzare i block cipher che trasformano un block cipher in uno stream cipher.

7 Modes Of Operation

I modes of operation sono tecniche utilizzate per combinare i block cipher ai messaggi con dimensione maggiori di un blocco.
I più utilizzati sono CBC (Counter Block Chaining), CTR (Counter Mode), CGB (Cipher Feedback Mode), OFB (Output Feedback Mode). I più avanzati sono GCM (Galois Counter Mode), OCB (Offset Codebook Mode).

8 CBC- Cipher Block Chaining

Il CBC (Cipher Block Cipher) è una tecnica di ENC e DEC per plaintext di dimensione maggiore di quella di un blocco. In particolare, vengono risulte le principali problmeatiche di ECB.Infatti, nella fase di encryption si prende un IV generato da un TRNG viene XORato con il plaintext, si applica il PRF e il suo output sarà il ciphertext che fungerà da IV per il blocco successivo. Si invia come ciphertext l'IV affiancato da tutti i blocchi cifrati. Nella fase di decryption si prende l'intero ciphertext, lo si suddivide in blocchi. Ora, partendo dal primo blocco si applica la funzione inversa di PRF che verrà XORato con l'IV per produrre il plaintext;il cipher del blocco precedente verrà utizzato come IV del blocco successivo per produrre il plaintext. Le problematiche di questo protocollo è che necessita di due circuiti per essere attuato (uno per ENC e uno per DEC), non è totalmente paralizzabile ma lo è solamente la fase di DEC. Quindi, la decript è molto veloce (a discapito dell'ENC).
Questa tecnica viene detta chaining poiché si lega il risultato precedente a quello successivo.
Questa caratteristica, tuttavia, presenta delle problematiche dato che il PRF soffre del cosidetto problema degli shor cycle:presa una permutazione dei blocchi, in base al punto di partenza con cui si applica la catena di PRF, può presentare dei cicli che rappresentano un problema di sicurezza dato che rivelano delle informazioni sul messaggio cifrato.
Di questo problema ne soffrono anche delle versioni parallelizzate e/o simili a CBC dette OFB (Output Feedback Mode) e CFB (Cipher Feedback Mode).

In CFB, la fase di encryption è così composta:si prende l'IV, si applica la funzione PRF e a questo punto il suo risultato viene XORato con il plaintext, il risultato dello XOR rappresenta l'IV per il blocco successivo e così via;per la fase di decryption, si prende l'IV, si applica la funzione PRF e se ne effettua lo XOR con il primo blocco del ciphertext, che in seguito rappresenterà l'IV per il blocco successivo, per ottenere il plaintext. Questa costruzione, come quella di OFB, ha la caratteristica di usare la funzione PRF sempre in avanti sia in ENC che DEC riducendo così i circuiti che vengono utilizzati per applicare questo algoritmo. Inoltre, questa costruzione non risulta parallelizzabile, ma ha una fase di ENC lenta mentre la DEC veloce.
La seconda alternativa è quella parallelizzabile sia in ENC che DEC detta OFB (Output Feedback Mode). Nell'encryption si prende l'IV si applica PRF, che fungerà da IV per il blocco successivo, si applica lo XOR con il plaintext per ottenere il ciphertext. Nella fase di decryption si prende l'IV si applica PRF. Il suo risultato sarà l'IV del blocco successivo, ma prima verrà XORato con il ciphertxt per produrre il plaintext.

9 Counter Mode-CTR

Il CTR è un modes of operation che prevede l'utilizzo di un contatore incrementale che viene utilizzato come IV. In particolare si iniziliazza il contatore, entra nel blocco PRP e il suo output viene XORato con il plaintext. A differenza da CBC e derivati, qui ogni blocco viene cifrato e decifrato in maniera singola rendendo più efficiente il sistema dato che non si necessita più di decifrare l'intero pacchetto per verificare l'autenticità del pacchetto in mio possesso.
In un'applicazione reale, si aggiunge sempre una parte di randomicità affiancando a questo contatore 96 su 128 bit una sequanza di bit casuale. Inoltre, questo procedimento equivale all'effettiva trasformazione dei block cipher in stream cipher.

10 BEAST Attack e Chosen Bondary Attack

Il BEAST attack è un attacco ai block cipher ed in particolare al CBC (Cipher Block Chaining) che sfrutta la problematica di short cycle e quindi della predicibilità dell'IV. In particolare, supponiamo che l'attacker possa predirre l'IV del blocco successivo, che abbiamo visto il ciphertext del blocco precedente e che possa far cifrare un messaggio a sua scelta.
Se CBC fosse CPA, allora la predicibilità dell'IV non dovrebbe nuocere alla sicurezza e soprattutto condizione necessaria e sufficiente per essere CPA è che l'IV non si ripeta mai e che non deve essere predicibile.
L'attacco procede come segue, l'attacker conosce il ciphertext del blocco precedente e ipotizza che nel campo password del ciphertext vi sia una password fra due scelte. A questo punto, l'attacker prende il ciphertext del blocco precedente e lo mette in XOR con l'IV e con la password che ipotizza esserci e ci applica

PRP. Questa azione, nel momento dell'encrypt fa cancellare l'IV predetto e se nel ciphertext finale vi è il ciphertext del blocco che voglio indovinare allora conosco la password del plaintext.
Nonostante questo attacco sia potente poiché mi permette di conoscere l'intero blocco che vogliamo attaccare non è molto pratico dato che deve essere flessibile alle tecnologie utilizzate per confincere l'implementazione a cifrare un messaggio a nostra scelta, bisogna avere un agente in esecuzione nel browser e si necessita la possibilità di effettuare injection di plaintext all'interno del messaggio. Anche se un attack riuscisse ad avere tutte questi poteri dovrebbe essere in grado di effettuare un brute force attack per conoscere il blocco da decifrare.
Una sua altertaniva, prende il nome di chosen boundary attack. Questi tipi di attacchi sono lineari nella dimensione del messaggio e permettono di scoprire un byte/bit alla volta del plaintex. Infatit, si fa corrispondere l'ultimo bit che voglio decifrare con la fine del blocco e facendo cifrare un messaggio a mia scelta si effettua un brute force attack per capire un byte alla volta partendo dall'ultimo blocco.

11 MAC-Then-Encrypt, Padding Oracle Attack

I Padding Oracle Attack sono un gruppo di euristiche di attacchi che si possono effettuare ogni qual volta che si utilizza il padding in un protocollo di sicurezza e si basano sulla possibilità di indovinare la lunghezza del padding.
In TLS, il padding del messaggio viene effettuato tramite la specifica PKCS 7 che consiste di inserire alla fine dei dati da inviare un byte contenente la lunghezza del padding del messaggio e i byte precedenti, fino al raggiungimento della lunghezza di padding, vengo riempiti con il lavore della lunghezza di pagging. Inoltre, vi è la possibilità di aggiungere blocchi interi di padding.
La scelta di utilizzare questo tipo di padding può rivelarsi utile dato che tramite esso si possono rilevare e scartare pacchetti malformati sia causati da errori di connessione e sia causati da attacchi alla comunicazione.
In TLS, si scelse di utilizzare CBC (Cipher Block Chaining) come algoritmo di cifratura e le operazioni che vengono effettuate sono MAC-then-Encrypt nel seguente ordine:prima si inizia a decifrare il messaggio se la lunghezza del messaggio non è un multiplo della dimansione del blocco o se non ha un padding corretto, allora si invia un alert decryption failed;inoltre, se queste due fasi sono andate a buon fine si fa il check di integrità sul messaggio decifrato e si ritorna un messaggio BAD_MAC,
Questo sistema di notifica potrebbe sembrare utile, ma a livello crittografico no poiché si danno informazioni all'attaccante e bisogna stare anche attenti all'implementazione di questi algoritmi poiché vi si potrebbe lasciare dei side channel aperti che consentirebbero di rompere le patch a implementazioni non corrette. Il Padding Oracle Attack si basa su questo.
Ipotizziamo che un attaccante vuole conoscere l'ultimo byte del messaggio e la sua abilità è quella di poter modificare il ciphertext e di sottoporre il nuovo ciphertext ad un oracolo che mi dice se il padding è ben formato oppure se ho

un BAD_MAC. Nel caso in cui si utilizzi CBC, devo modificare l'ultimo byte del blocco del ciphertext precedente a quello che voglio indovinare, scartando tutti blocchi successivi a quello che voglio indovinare, e per fare ciò devo modificarlo effettuando lo XOR tra il ciphertext che conosco, il byte che penso sia il pt e un padding valido secondo lo standard di TLS. A questo punto, invio ad un oracolo il mio nuovo ciphertext modificato:se mi risponde con un alert decryption error allora il padding non è stato ben formato e non ho invodinato il byte;altrimenti riceverò un BAD_MAC alert che mi indica che il padding è valido (sono riuscito a indovinare l'ultimo byte e il ciphertext presenta il padding da me voluto). Iterando questo procedimento con i restanti byte e mantenendo le modifiche dei guess precedenti, posso risalire a tutto il blocco.
Vi è anche la possibilità di indovinare un intero blocco in una passata prestando attenzione a effettuare lo XOR dell'intero blocco con un padding valido e un guess valido.
Questa tipologia di attacco rientra nelle cosiddette chosen ciphertext attack CCA:schemi di attacco in cui all'attacker viene data la facoltà di scegliere e cifrare un ciphertext di sua scelta.