Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Matematica

PROTOCOLLO DI AUTENTICAZIONE STS E IKE

Tesi di laurea in Algoritmi della Teoria dei Numeri e Crittografia

Relatore:
Chiar.mo Prof.
Davide Aliffi

Presentata da:
Laura Gurioli

Sessione Unica
Anno Accademico 2016/2017

Introduzione

In crittografia l'insieme delle procedure che permettono lo scambio di informazioni tra diverse entità è definito mediante protocolli, ciascuno dei quali ha precise finalità.
L'oggetto di studio di questa tesi è il protocollo di autenticazione Station To Station, una variazione del protocollo di Diffie-Hellman che, oltre ad ottenere una chiave segreta condivisa, permette a due soggetti di essere certi dell'identità del loro interlocutore.
La forza di questo protocollo è proprio nell'autenticazione, grazie alla quale riesce a resistere al cosiddetto attacco "Man in the middle".
Nel primo capitolo vengono definiti alcuni concetti base necessari a comprendere i contenuti dei capitoli successivi.
Tra questi la differenza tra crittografia simmetrica e asimmetrica, il logaritmo discreto, che rappresenta una componente fondamentale nella struttura del protocollo di Diffie-Hellman, e la firma digitale.
Il capitolo successivo tratta del protocollo di Diffie-Hellman, il primo sistema asimmetrico che permette lo scambio di una chiave tra due soggetti senza l'utilizzo di un canale privato.
Questo sistema, seppure abbastanza sicuro, è vulnerabile all'attacco "Man in the middle" che permette all'avversario di inserirsi "in mezzo" alla conversazione e controllarla all'insaputa dei due interlocutori.
E' proprio per far fronte alla minaccia dell'attacco "Man in the middle" che, nel 1987, il protocollo di Diffie-Hellman si è evoluto nel protocollo STS, che viene descritto nel terzo capitolo.

In questo protocollo ognuno dei due interlocutori è certo dell'identità dell'altro grazie all'inserimento della firma digitale.
STS ha numerose proprietà ma anche qualche limite, come ad esempio l'attacco di Lowe.
Infine, nell'ultimo capitolo, viene illustrata un'applicazione del protocollo STS, il sistema IKE per l'autenticazione e lo scambio delle chiavi su Internet.
Partendo dalla descrizione della stuttura a livelli dello standard ISO/OSI per la trasmissione di messaggi in rete, arriveremo a descrivere il protocollo IP che si occupa dell'indirizzamento dei pacchetti che viaggiano attraverso la rete e vedremo come questo protocollo può acquisire maggiore sicurezza.
A questo scopo è nato IPSec, un sistema che permette di aggiungere una protezione crittografica a IP per renderlo più sicuro.
Come ultima cosa verrà descritto il funzionamento di IKE che è attualmente lo standard per lo scambio delle chiavi e l'autenticazione delle parti per IPSec.

Indice

Capitolo 1

Nozioni preliminari

1.1 Crittografia simmetrica e asimmetrica

La crittografia si divide in due importanti classi: crittografia simmetrica e crittografia asimmetrica. Per lo studio di questa tesi è necessario concentrarsi in particolare sul secondo tipo.

1.1.1 Crittografia simmetrica

La crittografia simmetrica si basa sull'esistenza di un'unica chiave che viene mantenuta segreta; essa è utilizzata sia per cifrare che per decifrare il messaggio, grazie a un algoritmo di cifratura facilmente reversibile.
Il grande limite di questo sistema è la necessità di avere a disposizione un canale sicuro, attraverso il quale le parti possono scambiarsi la chiave senza che un eventuale avversario riesca ad intercettarla.
La crittografia a chiave simmetrica è la forma più antica di crittografia e fino al secolo scorso l'unica forma esistente.
I primi ad utilizzare forme di crittografia simmetrica furono gli egizi più di 4500 anni fa, successivamente si svilupparono tecniche sempre migliori tra cui ricordiamo i cifrari classici (quello di Cesare ne è un esempio), fino ad arrivere a sistemi più moderni come Rijndael.

1.1.2 Crittografia asimmetrica

La crittografia asimmetrica è detta anche crittografia a chiave pubblica in quanto prevede l'utilizzo di una coppia di chiavi (una pubblica e una segreta) per ciascuna persona coinvolta nella conversazione.
A differenza della crittografia simmetrica tutti gli scambi si svolgono su un canale pubblico, eliminando la necessità di avere a disposizione un canale sicuro per lo scambio della chiave. In questo modo due persone possono comunicare in modo riservato anche se non si sono mai incontrate.
La chiave pubblica viene condivisa mentre la chiave privata rimane personale e segreta.
La sicurezza di questi sistemi è dovuta unicamente alla difficoltà computazionale di alcuni problemi matematici per cui non sono conosciuti algoritmi efficienti di risoluzione, come ad esempio la fattorizzazione di un intero o il logaritmo discreto.
La forza di un sistema di crittografia a chiave pubblica si basa sulla difficoltà di determinare la chiave privata corrispondente alla chiave pubblica.
Questo tipo di crittografia è però computazionalmente pesante, perciò viene solitamente abbinata alla crittografia simmetrica. Con il primo sistema viene scambiato un piccolo blocco di dati, che viene poi utilizzato come chiave di sessione del sistema simmetrico per cifrare testi più lunghi.

1.2 Logaritmo discreto

Definizione 1.1.
Sia G un gruppo ciclico di ordine n con operazione di gruppo $*$ e sia g un generatore di questo gruppo.
Ogni elemento x di G può essere scritto nella forma $x = g^k$ per un certo intero k.
Chiamiamo k il logaritmo discreto di x, ovvero $log_g x$.

Dato un gruppo G, la difficoltà del calcolo del logaritmo discreto di un suo elemento dipende dalla grandezza dell'ordine del gruppo ma anche dai suoi fattori primi.
Il logaritmo discreto è comunque un'operazione piuttosto complessa e ad oggi non sono noti algoritmi efficienti per la sua risoluzione; per questo motivo risulta particolarmente adatto ad essere usato in crittografia.
El Gamal e Diffie-Hellman sono esempi di metodi basati sul logaritmo discreto, entrambi utilizzano i gruppi ciclici $\mathbb{Z}_p^*$, con p primo.
Più recentemente il logaritmo discreto ha trovato applicazione anche nel campo della crittografia sulle curve ellittiche.

Esempio 1.1.
Sia p un numero primo e consideriamo il caso del gruppo moltiplicativo degli interi modulo p, $\mathbb{Z}_p^*$.

$$\mathbb{Z}_p^* = \{1, \ldots, p-1\}$$

Ad esempio consideriamo $\mathbb{Z}_{13}^*$, con lo scopo di determinare $log_5 8 \bmod 13$.
Dato $5^k \equiv 8 \mod 13$, l'obiettivo è quello di calcolare il più piccolo k che verifica l'uguaglianza.
In questo esempio banale è sufficiente trovare la potenza discreta 5^3. Per prima cosa calcoliamo $5^3 = 125$ e dividiamo 125 per 13, ottenendo 9 con il resto di 8; perciò nel gruppo $\mathbb{Z}_{13}^*$ si ha $5^3 = 8$.
Quindi lo scopo è raggiunto: $log_5 8 = 3 \bmod 13$.

1.3 Firma digitale

La firma digitale è una componente crittografica fondamentale nei sistemi asimmetrici.

Essa rappresenta l'equivalente informatico di una tradizionale firma autografa apposta su carta, e serve a dimostrare l'autenticità di un messaggio inviato attraverso un canale insicuro.
In particolare le sue funzioni sono:

- **Autenticazione:** chi riceve il messaggio deve essere sicuro dell'identità del mittente
- **Non ripudiabilità:** il mittente non può affermare di non aver inviato il messaggio
- **Integrità:** la firma garantisce che il messaggio non sia stato modificato lungo il tragitto

Osservazione 1.
La firma digitale non può essere una semplice digitalizzazione della firma cartacea, poiché sarebbe molto facile per l'avversario estrarre dal documento digitale la parte contenente la firma e copiarla su un altro documento.
Quindi, a differenza della firma manuale, la firma digitale deve dipendere non solo dal mittente ma anche dal documento a cui è legata, per essere inscindibile da questo ed evitare che possa essere falsificata tramite un attacco elementare.

Osservazione 2.
Ciascun utente che desideri utilizzare la firma digitale deve essere in possesso di un certificato digitale, che attesti l'identità del soggetto e garantisca la corrispondenza biunivoca tra la chiave privata di firma e il suo possessore.
Un terzo soggetto, il certificatore, verifica e garantisce l'affidabilità dei dati e infine li pubblica in un registro in modo che le parti possano riporre piena fiducia nei certificati in esso contenuti.

1.3.1 Procedura generale di firma

Chiamiamo Alice (A) il firmatario e Bob (B) il verificatore della firma.

Terminologia:

- M è l'insieme di messaggi che possono essere firmati
- S è l'insieme delle firme, che solitamente sono stringhe binarie di lunghezza fissa
- S_A è una funzione che associa ad ogni messaggio dell'insieme M una firma dell'insieme S. La trasformazione S_A è tenuta segreta da A e verrà utilizzata per creare firme per i messaggi dell'insieme M
- V_A è un'applicazione dall'insieme $M \times S$ all'insieme $\{true, false\}$. V_A ha funzione di verifica per le firme di A ed è conosciuta pubblicamente

Procedura di firma:

A crea la firma per un messaggio $m \in M$ in questo modo:

1. Calcola $s = S_A(m)$
2. Trasmette la coppia (m, s) a B.

Procedura di verifica:

B riceve la coppia (m, s) e deve verificare che una firma s su un messaggio m sia stata creata da A. Esegue le seguenti operazioni:

1. Ottiene la funzione di verifica V_A di A
2. Calcola $u = V_A(m, s)$
3. La firma è stata creata da A se $u = true$ mentre se $u = false$ la firma viene rifiutata.

Ci sono alcune **proprietà** che le trasformazioni di firma e verifica devono soddisfare:

- s è una firma valida di A sul messaggio m se e solo se $V_A(m, s) = true$
- È computazionabilmente impossibile per qualsiasi soggetto diverso da A trovare, per qualsiasi $m \in M$, un $s \in S$ tale che $V_A(m, s) = true$

Ci sono alcuni metodi che si ritiene soddisfino le due proprietà, tuttavia nessuno ha ancora dimostrato formalmente l'esistenza di sistemi di firma digitale che soddisfino la seconda condizione (sebbene l'esistenza sia ritenuta vera).

Capitolo 2

Scambio delle chiavi di Diffie-Hellman

Lo scambio delle chiavi di Diffie-Hellman è un protocollo crittografico che consente a due interlocutori, Alice e Bob, di ottenere la stessa chiave segreta senza la necessità di disporre di un canale di comunicazione sicuro.
Questo protocollo, pubblicato per la prima volta nel 1976, fu il primo a consentire a due entità di scambiarsi un'informazione segreta utilizzando interamente un canale di comunicazione pubblico.
Non è necessario, infatti, che le due parti si siano scambiate informazioni o si siano incontrate in precedenza.
La sicurezza di questo sistema è dovuta al fatto che il calcolo del logaritmo discreto è computazionalmente proibitivo per numeri grandi e ad oggi non sono noti algoritmi efficienti per la risoluzione di questo problema.
La chiave ottenuta grazie a questo scambio, può essere successivamente utilizzata per cifrare messaggi con metodi di crittografia simmetrica.

2.1 Descrizione

Andiamo ora a descrivere i passaggi del metodo, supponendo che Alice e Bob vogliano scambiarsi una chiave.

- Alice e Bob si accordano su un numero primo p e un generatore g del gruppo moltiplicativo degli interi modulo p. Questi valori vengono resi pubblici e chiunque può accedervi
- Alice genera un numero casuale a con $1 \leq a \leq p-2$, calcola $A = g^a$ mod p e lo invia a Bob attraverso il canale insicuro
- Bob genera un numero casuale b con $1 \leq b \leq p-2$ e calcola $B = g^b$ mod p e lo invia ad Alice
- Alice riceve B calcola la chiave segreta condivisa $K_A = B^a$ mod p
- Bob riceve A e calcola la chiave segreta condivisa $K_B = A^b$ mod p

Si noti che i valori calcolati sono gli stessi, infatti:

$$K_A = B^a = (g^b)^a = (g^a)^b = A^b = K_B$$

Alice e Bob hanno così ottenuto una chiave comune che possono sfruttare in altri algoritmi crittografici simmetrici.
Se un avversario dovesse assistere a tutto lo scambio non sarebbe comunque in grado di risalire alla chiave segreta in quanto l'operazione del logaritmo discreto è molto onerosa dal punto di vista computazionale e richiede un tempo sub-esponenziale.
Questo sistema è però molto vulnerabile ad attacchi di tipo "man in the middle".

2.2 Attacco Man in the middle

L'attacco "Man in the middle", letteralmente "uomo nel mezzo", è un attacco mediante il quale un avversario, che chiameremo Eva, riesce ad im-

personare uno o entrambi gli interlocutori riuscendo così ad ottenere informazioni riservate.
Con questo attacco Eva può segretamente ritrasmettere o alterare la comunicazione tra due soggetti che credono di comunicare direttamente tra di loro. Vediamo ora questo tipo di attacco nello scambio delle chiavi di Diffie-Hellman, supponendo che Alice e Bob vogliano scambiarsi dei messaggi e che Eva voglia spiare o alterare la conversazione, inviando alle due parti dei messaggi contraffatti.

- Vengono scelti un numero primo p e un generatore g del gruppo moltiplicativo degli interi modulo p. Entrambi i valori vengono resi pubblici.
- Alice genera la sua chiave segreta a $(1 \leq a \leq p-2)$, calcola $A = g^a$ e lo invia a Bob.
- Bob genera a sua volta un numero casuale b $(1 \leq b \leq p-2)$,calcola $B = g^b$ e lo invia ad Alice.
- Eva, con lo scopo di introdursi nella conversazione e controllarla, genera a sua volta un numero e $(1 \leq e \leq p-2)$ e calcola $E = g^e$.
- Eva intercetta A che era destinato a Bob e lo rimpiazza con E fingendosi Alice.
- Eva intercetta B che era destinato ad Alice e lo rimpiazza con E impersonando Bob.
- Alice riceve E e credendo che sia stato inviato da Bob calcola la chiave $K_A = E^a = (g^e)^a = A^e$.
- Bob riceve E e credendo che sia stato inviato da Alice calcola la chiave $K_B = E^b = (g^e)^b = B^e$.
- Eva calcola sia la chiave $K_A = A^e$, con cui può conversare con Alice fingendosi Bob, sia la chiave $K_B = B^e$ che le servirà per scambiare messaggi con Bob impersonando Alice.

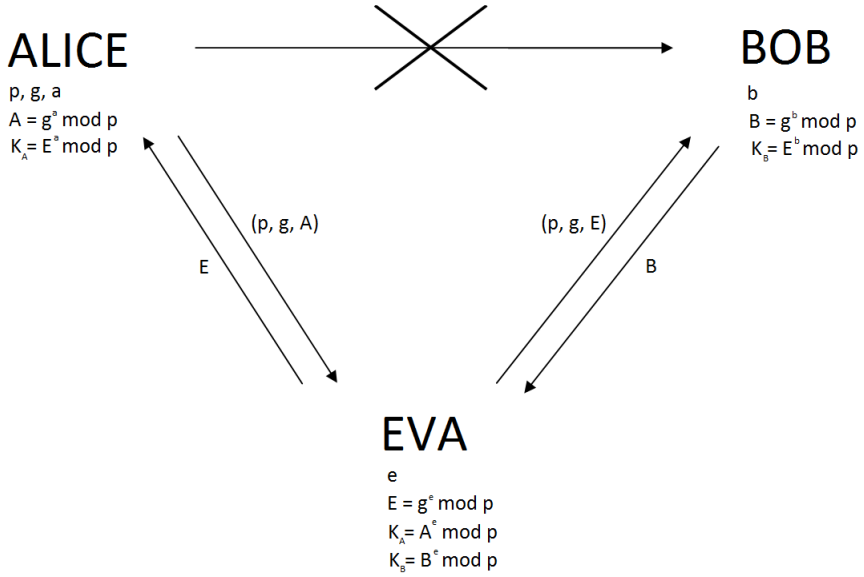


Figura 2.1: Attacco Man in the middle

A questo punto A e B credono di avere una chiave segreta condivisa, ma in realtà non sono in comunicazione tra loro.
Eva invece ha a disposizione sia una chiave segreta condivisa con Alice (K_A) che una chiave segreta condivisa con Bob (K_B).
Ora Eva controlla la conversazione e può scegliere se modificare a suo piacere i messaggi che Alice e Bob pensano di scambiarsi o se limitarsi a spiarne il contenuto riinviandoli al destinatario inalterati.
L'attacco Man in the middle al protocollo Diffie-Hellman può essere evitato con l'uso delle firme digitali e dei certificati a chiave pubblica.

Capitolo 3

Protocollo di autenticazione STS

Il protocollo STS (Station To Station) è una variazione del protocollo base di Diffie-Hellman presentato per la prima volta nel 1987 nel contesto della sicurezza ISDN[1].
Esso permette la creazione di una chiave segreta condivisa tra due soggetti, inoltre consente di essere certi dell'autenticità delle parti con cui avviene lo scambio della chiave.
Il metodo utilizza la firma digitale, che garantisce sicurezza contro gli attacchi man in the middle, diversamente dal classico Diffie-Hellman che invece è vulnerabile ad attacchi di questo tipo.
Oltre a questo, il protocollo STS fornisce la Perfect Forward Secrecy (segretezza in avanti); grazie a questa proprietà se una chiave segreta viene compromessa le chiavi di sessione generate in precedenza rimangono riservate.
Il protocollo STS ha un ruolo influente nel campo dell'autenticazione, inoltre è una delle basi per il protocollo di autenticazione standard per la sicurezza Internet chiamato "Internet Key Exchange (IKE)".

[1]Integrated Services Digital Network: rete di telecomunicazioni digitale che da supporto a molti servizi di voce e dati.

3.1 Descrizione

Prima di iniziare il protocollo, è necessario generare i seguenti dati:

- Una coppia di chiavi (una pubblica e una segreta) per la firma asimmetrica per ciascuna delle parti, necessaria per l'autenticazione. La parte pubblica di questa coppia di chiavi può essere condivisa prima della creazione della sessione.
- Un numero primo p e un generatore g del gruppo ciclico degli interi modulo p. Questi parametri sono resi pubblici.

Supponendo che tutti i dati di configurazione siano stati condivisi, il protocollo STS procede come segue. Se non è possibile completare un passaggio, il protocollo si arresta immediatamente.

- Alice genera un numero casuale x, calcola g^x e lo invia a Bob.
- Bob genera un numero casuale y e calcola g^y. Inoltre determina la chiave segreta condivisa $K = (g^x)^y$. Successivamente concatena gli esponenziali (g^y, g^x) (l'ordine è importante), li firma usando la sua chiave privata, e poi cifra la firma con K. Infine invia il testo cifrato insieme a g^y ad Alice.
- Alice calcola la chiave segreta condivisa $K = (g^y)^x$, decifra e verifica la firma di Bob utilizzando la sua chiave pubblica. Successivamente concatena gli esponenziali (g^x, g^y), li firma usando la sua chiave privata e poi cifra la firma con K. Infine invia il testo cifrato a Bob.
- Bob decifra e verifica la firma di Alice usando la sua chiave pubblica.

Alice e Bob sono ora certi dell'identità del loro interlocutore e condividono la stessa chiave segreta K che gli sarà utile per cifrare comunicazioni successive.
Il protocollo può essere sintetizzato nei seguenti passaggi, dove omettiamo lo scambio dei certificati:

1. $A \xrightarrow{g^x} B$
2. $A \xleftarrow{g^y, E_K(S_B(g^y, g^x))} B$
3. $A \xrightarrow{E_K(S_A(g^x, g^y))} B$

3.2 Proprietà

Il protocollo STS gode di 4 proprietà:

1. **Autenticazione reciproca delle due entità**
   Ciascuna delle due parti è certa dell'identità del proprio interlocutore. Inoltre, il fatto che la chiave di sessione venga usata già nello scambio dei messaggi all'interno del protocollo, dovrebbe garantire contro attacchi di "replay", in cui Eva ritrasmette messaggi intercettati in precedenza. In altre parole Alice e Bob si convincono di parlare con un interlocutore "online".

2. **Chiave reciprocamente autenticata**
   Questa proprietà deriva dal protocollo di Diffie-Hellman, la validità della chiave concordata è garantita se ogni parte ha selezionato il suo esponente casuale correttamente.

3. **Convalidazione reciproca della chiave**
   Al termine dello scambio entrambe le parti hanno visto che il loro interlocutore ha utilizzato la chiave condivisa per crittografare il materiale dell'accordo.

4. **Perfect Forward Secrecy**
   Questa è una proprietà molto importante in un protocollo.
   Grazie a questa proprietà, se ad un certo momento la chiave privata di Alice o di Bob (o entrambe) viene compromessa e di conseguenza Eva riesce ad ottenere la chiave condivisa K, la sicurezza di qualsiasi sessione stabilita prima di quel momento non sarà intaccata.

Dal momento che la chiave di sessione è una funzione unidirezionale di due chiavi segrete, che cambieranno per ogni conversazione, compromettere entrambe le chiavi private non può avere alcun effetto sulla segretezza delle chiavi condivise delle sessioni precedenti ma renderà possibili attacchi Man in the middle successivi.

3.3 Protocollo STS semplificato

Il protocollo STS è stato semplificato in un protocollo chiamato "Authentication-only" che è in parte uguale al protocollo proposto dall'ISO "three-way authentication protocol" ma con un'importante differenza: i messaggi firmati non contengono l'identità dei soggetti che partecipano al protocollo.

I passi dell'algoritmo sono i seguenti:

1. $A \xrightarrow{R_A} B$
2. $A \xleftarrow{R_B, sign_B(R_B, R_A)} B$
3. $A \xrightarrow{sign_A(R_A, R_B)} B$

Anche in questo caso A e B dimostrano la propria identità firmando con la propria chiave segreta i nonce R_A e R_B[2]. Il protocollo STS semplificato è vulnerabile all'attacco "certificate-signature-replacement attack", dove l'avversario (Eva), che è un utente legittimo del sistema, e quindi dispone di una chiave pubblica certificata, attende che Alice inizi una conversazione.
In questo caso, Eva fa partire uno scambio di messaggi con Bob impersonando Alice e usando il suo nonce.
Dopo aver ricevuto la risposta di Bob, Eva sostituisce il certificato e la firma

[2]Nonce: numero casuale che viene utilizzato in modo unico nei protocolli di autenticazione per assicurarsi che i dati scambiati non possano essere utilizzati in attacchi di tipo replay nelle conversazioni successive.

di quest'ultimo con le proprie.
In questo modo riesce a persuadere Alice a firmare il nonce di Bob, permettendo a Eva di ingannarlo con successo.
Questo attacco, basato sulla sostituzione della firma, non può essere applicato al protocollo STS perché la crittografia utilizzata nella versione completa del protocollo impedisce a Eva di sostituire la firma di Bob.
Aggiungere l'identità dei partecipanti all'interno della firma costituisce effettivamente un metodo per evitare questo tipo di attacco, non è però l'unico modo per risolvere il problema.
In alcune applicazioni (ad esempio "Protocollo Internet Key Exchange (IKE)"), le identità dei partecipanti al protocollo possono essere omesse per ottenere una proprietà di privacy.

Ecco uno schema dell'attacco "certificate-signature-replacement attack":

1. $A \xrightarrow{R_A} E \xrightarrow{R_A} B$
2. $A \xleftarrow{R_B, sign_E(R_B, R_A)} E \xleftarrow{R_B, sign_B(R_B, R_A)} B$
3. $A \xrightarrow{sign_A(R_A, R_B)} E \xrightarrow{sign_A(R_A, R_B)} B$

3.4 Attacco di Lowe

Definizione 3.1.
Secondo gli autori del protocollo STS per definire uno scambio sicuro in un protocollo di autenticazione si può utilizzare la nozione di "matching records of runs".
"Matching records of runs" significa che ogni messaggio scambiato da un qualunque soggetto coinvolto nel protocollo viene registrato da ogni altro partecipante in ordine cronologico, facendo in modo che tutte le parti abbiano lo stesso registro.
Di conseguenza una conversazione risulta insicura se il registro dei partecipanti non coincide.

Vediamo come si svolge l'attacco di Lowe:

1. $A \xrightarrow{g^x} E \xrightarrow{g^x} B$
2. $A \xleftarrow{g^y, E_K(S_B(g^y, g^x))} E \xleftarrow{g^y, E_K(S_B(g^y, g^x))} B$
3. $A \xrightarrow{E_K(S_A(g^x, g^y))} E \qquad B$

In questo attacco Eva si relaziona con Bob usando la sua vera identità, mentre negli scambi con Alice finge di essere Bob.
Eva, dopo aver intercettato il messaggio g^x inviato da Alice e destinato a Bob, lo inoltra a quest'ultimo simulando di voler iniziare una sessione con lui e alla fine riesce a far accettare ad Alice la propria falsa identità come Bob.
Quello di Lowe è un attacco legittimo al protocollo STS, anche se il danno che può causare è molto limitato.
L'attacco di Lowe non è particolarmente dannoso in quanto:

1. Nello scambio tra Alice e Eva, nonostante la seconda sia riuscita ad ingannare la prima, non conosce la chiave di sessione condivisa e quindi non può sfruttarla facendosi credere Bob, per imbrogliare Alice in conversazioni future.

2. Nello scambio tra Eva e Bob, Eva non può completare lo scambio, e quindi l'attacco non ha successo.

Questo attacco è comunque legittimo per due ragioni:

1. Alice è convinta di conversare con Bob, questo è il risultato della copia effettuata da Eva di tutti i messaggi che Bob le ha inviato.
   Tuttavia, i messaggi registrati da Bob non corrispondono a quelli registrati da Alice (manca l'ultimo messaggio). Pertanto, l'attacco risulta insicuro secondo quanto definito dagli autori del protocollo STS, cioè l'autenticazione reciproca non riesce.

2. Eva inganna Alice facendole credere di interagire con Bob, le sue successive richieste di comunicazioni sicure con quest'ultimo però saranno

negate senza alcuna spiegazione finché Bob penserà di non essere mai stato in comunicazione con Alice.

L'attacco di Lowe può essere realmente una preoccupazione se, per esempio, Bob è un server centralizzato e subisce un attacco distribuito (attacco di massa lanciato dal team di Eva).

La mancanza di notifica dagli utenti finali è un grave problema, infatti il server riserverà risorse per molti utenti finali e la sua capacità di servirne altri calerà drasticamente.

Notiamo che questo attacco costa davvero poco a Eva perchè non utilizza certificati crittografici, a differenza di attacchi del tipo "denial-of-service" dove utilizza il suo vero nome per scambiare messaggi.

Se il protocollo viene modificato in modo da seguire il principio proposto da Abadi e Needham questo attacco può essere evitato.
Questo principio dice che è opportuno menzionare esplicitamente il nome del mittente e del destinatario all'interno di ogni messaggio.
Se applichiamo questo principio al protocollo STS, includendo nel messaggio le identità di entrambi i partecipanti alla conversazione, il messaggio inviato e firmato da Bob contiene il nome "Eva"; ciò impedisce a Eva di inoltrarlo ad Alice spacciandosi per Bob.
Anche questa volta aggiungere l'identità dei partecipanti all'interno della firma costituisce effettivamente un metodo per evitare questo tipo di attacco, anche se non l'unico.

Capitolo 4

Un'applicazione del protocollo STS: IKE

4.1 Il modello ISO/OSI

Le tecniche di interconnessione tra computer inizialmente erano reti "chiuse", ovvero capaci di mettere in comunicazione soltanto computer prodotti dallo stesso costruttore.
Negli anni 70 nasce il bisogno di avere sistemi informativi "aperti", meccanismi capaci di far lavorare assieme calcolatori eterogenei; sorge così la necessità di utilizzare standard condivisi.
Tali standard sono i modelli TCP/IP e ISO/OSI; entrambi hanno una struttura a livelli (layer) in cui ogni livello ha un ruolo preciso nella trasmissione dei dati.
Essi si occupano di suddividere l'insieme dei protocolli di rete, su cui si fondano tutte le comunicazioni effettuate attraverso Internet, in base alla loro funzione.
I livelli più alti sono più vicini all'utente e lavorano su dati più astratti, via via che i livelli si abbassano i dati vengono tradotti in forme più adatte ad essere manipolate ed infine trasmesse sul canale di comunicazione.
Il modello ISO/OSI prevede 7 livelli, mentre il modello TCP/IP ne prevede

solo 4. I livelli del modello OSI sono i seguenti:

1. **Livello Fisico**: definisce il modo in cui i dati sono fisicamente convertiti in impulsi elettrici.

2. **Livello Collegamento Dati**: ha il compito di sincronizzare i dati, controllare il loro flusso e riconoscere gli errori.

3. **Livello Rete**: gestisce l'indirizzamento dei pacchetti che viaggiano in rete. Questo livello utilizza il protocollo IP.

4. **Livello Trasporto**: si occupa del trasferimento dei dati, della loro divisione in pacchetti e degli eventuali errori di trasmissione. Uno dei protocolli usati in questo livello è il TCP.

5. **Livello Sessione**: gestisce le modalità di dialogo tra gli elaboratori attraverso l'apertura e la distruzione delle sessioni di comunicazione.

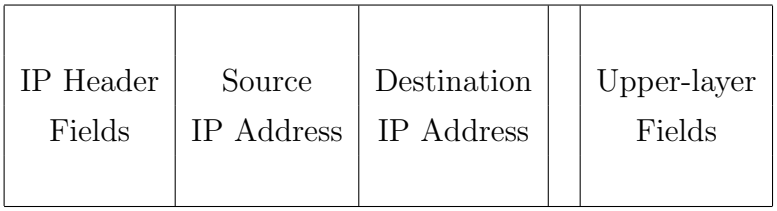
6. **Livello Presentazione**: definisce il formato standardizzato che i dati devono avere per poter essere manipolati da diversi dispositivi.

7. **Livello Applicazione**: rappresenta l'interfaccia con l'utente.

4.2 Il protocollo IP

Internet è una vastissima rete di dispositivi hardware, detti "nodi", ad ognuno dei quali è assegnato un indirizzo unico chiamato "indirizzo IP".
Il protocollo di rete IP (Internet Protocol) lavora al livello 3 nel modello ISO/OSI e si occupa dell'indirizzamento e l'instradamento dei dati tra dispositivi eterogenei.
I dati a questo livello sono suddivisi in pacchetti, ognuno di questi contiene 4 campi.
I primi tre contengono tutte le informazioni di overhead necessarie al trasferimento (indirizzo del mittente, indirizzo del ricevente, dati che riguardano

il pacchetto e l'assemblaggio con gli altri pacchetti) mentre l'ultimo include il protocollo del livello appena superiore (per esempio il TCP), ossia il segmento TCP, incapsulato nel pacchetto.



Per rendere la comunicazione riservata due nodi potrebbero decidere di applicare la crittografia end-to-end, ma solo il contenuto del messaggio nella quarta casella del "pacchetto IP" verrà crittografato, altrimenti l'indirizzamento non sarebbe possibile.
Se il protocollo IP non offre alcuna protezione, i primi tre campi ("IP header") non sono protetti ed è proprio la modifica di tali dati a costituire la principale causa di diversi attacchi.
Questi attacchi possono essere di vari tipi:

- spoofing: creazione di un pacchetto IP nel quale viene falsificato l'indirizzo IP del mittente
- sniffing: intercettazione passiva dei dati in transito
- dirottamento sessione: combinazione di spoofing e sniffing

Per far fronte a questo problema è stato introdotto l'IPSec.

4.3 Il protocollo IPSec

Il protocollo IPSec (Internet Protocol Security) consiste in una serie di processi standardardizzati per la sicurezza di IP e permette di aggiungere una protezione crittografica all'interno del pacchetto IP, rendendo il protocollo meno vulnerabile agli attacchi.

La famiglia di protocolli IPsec è composta da: **"Authentication Header"** (AH) e **"Encapsulated Security Payload"** (ESP).
L'autenticazione, di cui si occupa AH, è un servizio obbligatorio per IPSec mentre la riservatezza, garantita da ESP, è opzionale.
Il compito di AH, oltre alla verifica dell'autenticità del mittente, è quello di fornire un controllo di integrità e protezione contro i "replay attack".
Nella versione 6 di IP (IPv6) con la protezione IPSec, che lavora su blocchi di dati composti da multipli di 32 bit, il campo "Authentication Header" è posto tra l' "IP header" e l' "Upper-layer fields."

IP Header	Authentication Header	Upper-layer fields

Il campo "Authentication Header" è a sua volta strutturato come nella tabella seguente:

Next Header (8 bits)	Payload Lenght (8 bits)	Reserved for future use (16 bits)
Security Parameters Index (SPI, 32 bits)		
Sequence Number Field (32 bits)		
Authentication Data (blocchi di multipli di 32 bits)		

La riservatezza nel protocollo IPSec viene garantita grazie alla parte chiamata "Encapsulating Security Payload" (ESP) che viene aggiunta al pacchetto IP.
Il campo ESP, che è un servizio opzionale, viene eventualmente posizionato di seguito al campo AH.
Il formato di ESP è il seguente:

Security Parameters Index (SPI, 32 bits)	
Sequence Number Field (32 bits)	
Playload Data (blocchi di multipli di 32 bits)	
Padding (0-255 bytes)	
Pad Lenght (8 bits)	Next Header (8 bits)
Authentication Data (blocchi di multipli di 32 bits)	

Un concetto fondamentale per IPSec è quello di "Security Association" (SA). Un SA è univocamente definito da una terna:

$$(SPI, "IPDestinationAddress", "ServiceIdentifier")$$

dove "Service Identifier" può identificare o AH o ESP.

Due nodi che desiderano comunicare tramite IPSec devono necessariamente negoziare una SA (per l'autenticazione) o, eventualmente, due SA (per l'autenticazione e la riservatezza) oltre alle chiavi crittografiche segrete da condividere tra i due nodi per poter calcolare le protezioni crittografiche. La negoziazione è ottenuta utilizzando il protocollo Internet Key Exchange Protocol che introdurremo ora.

4.4 Il protocollo IKE

IKE (Internet Key Exchange) è l'attuale standard IETF per lo scambio delle chiavi e l'autenticazione delle parti per IPSec.
IKE è una suite di protocolli di autenticazione, ognuno dei quali usa parti dei protocolli "Oakley", "SKEME" e "ISAKMP".
Oakley descrive una serie di scambi della chiave e fornisce dettagli sui servizi forniti da ciascuno scambio.
SKEME descrive una tecnica per lo scambio di chiavi autenticate che supporta la negabilità e un rapido aggiornamento della chiave.

ISAKMP (Internet Security Association and Key Management Protocol) definisce le procedure e il formato dei pacchetti per la gestione delle SA, oltre che per lo scambio e l'autenticazione delle chiavi.
IKE è composto da due fasi:

- **Fase 1** In questa prima fase (ed è la parte che più ci interessa) si cerca di ottenere l'autenticazione reciproca, stabilendo una chiave di sessione condivisa utilizzata negli scambi della Fase 1.
  Se necessario la chiave può anche essere utilizzata per proteggere gli scambi nella Fase 2 e per garantire la comunicazione con il livello superiore.

- **Fase 2** Dopo gli scambi della Fase 1 può avvenire un certo numero di scambi della fase 2, che è spesso definita come "Modalità rapida".
  Gli scambi avvengono attraverso la chiave di sessione condivisa concordata nella Fase 1.
  Il motivo che porta ad avere un certo numero di scambi della Fase 2 è l'esigenza di poter impostare connessioni multiple con proprietà di sicurezza diverse.

4.4.1 Fase 1

Ci sono otto varianti della Fase 1 di IKE, questo perchè esistono quattro tipi di chiavi (chiave simmetrica pre-condivisa, chiave di cifratura pubblica vecchio stile, chiave di cifratura pubblica nuovo stile e chiave pubblica di verifica della firma) e per ogni tipologia di chiave ci sono due tipi di scambi: una "modalità standard" e una "modalità aggressiva".
Ogni "modalità standard" è composta da 6 scambi di messaggi, di cui 3 sono inviati da un mittente (A) a un ricevente (B) e 3 da B a A.
Una "modalità standard" è sempre obbligatoria, due utenti non possono utilizzare una "modalità aggressiva" senza prima aver effettuato una "modalità standard".
Ogni "modalità aggressiva" prevede lo scambio di soltanto 3 messaggi: A

invia un messaggio a B, B risponde e in fine A invia un ultimo messaggio per terminare lo scambio.[1]
La "modalità aggressiva" è facoltativa e può essere omessa.
Per la fase 1 di IKE descriviamo e analizziamo soltanto le modalità basate sulla firma digitale.

Signature-based IKE Phase 1 Main Mode

"Signature-based IKE Phase 1 Main Mode" è nato sotto l'influenza di diversi protocolli ma le sue vere radici risalgono a due in particolare: STS e il protocollo proposto da Krawczyk, denominato SIGMA.
Esso è costituito da 6 scambi, nei primi 2 A manda a B HDR_A e SA_A, e B risponde con HDR_B e SA_B.
HDR_A e HDR_B sono pacchetti di dati di intestazione rispettivamente di A e B; in essi sono contenuti i "cookies" C_A e C_B; questi ultimi sono stringhe di testo che servono a mantenere le informazioni sullo stato di esecuzione delle due parti.
SA_A e SA_B sono invece due Security Associations che servono a negoziare i parametri da utilizzare nel protocollo corrente, come algoritmi di crittografia, algoritmi di firma, funzioni pseudo-casuali per i messaggi di hashing da firmare ecc.
In particolare SA_A specifica le modalità di protezione che A vorrebbe utilizzare mentre SA_B specifica quelle scelte da B.
A è libero di proporre opzioni multiple, mentre B deve rispondere con una sola scelta.
La seconda coppia di messaggi non è altro che lo scambio delle chiavi Diffie-Hellman.
Negli ultimi due messaggi, gli algoritmi per la crittografia, la firma e le funzioni pseudo-casuali per i messaggi di hashing da firmare sono quelli concordati

[1] Sembra che, in generale, non sia possibile un'autenticazione reciproca con meno di 3 messaggi

nelle SA.

Ecco uno schema della Signature-based IKE Phase 1 Main Mode:

1. $A \xrightarrow{HDR_A, SA_A} B$
2. $A \xleftarrow{HDR_B, SA_B} B$
3. $A \xrightarrow{HDR_A, g^x, N_A} B$
4. $A \xleftarrow{HDR_B, g^y, N_B} B$
5. $A \xrightarrow{HDR_A, \{ID_A, Cert_A, Sig_A\}g^{xy}} B$
6. $A \xleftarrow{HDR_B, \{ID_B, Cert_B, Sig_B\}g^{xy}} B$

Sig_A e Sig_B sono le firme create rispettivamente da A e B, mentre N_A N_B sono i loro nonce. Se chiamiamo prf_1 e prf_2 le funzioni pseudo casuali concordate nelle SA, i messaggi firmati saranno:

$$M_A = prf_1(prf_2(N_A|N_B|g^{xy})|g^x|g^y|C_A|C_B|SA_A|ID_A)$$

$$M_B = prf_1(prf_2(N_A|N_B|g^{xy})|g^y|g^x|C_B|C_A|SA_B|ID_B)$$

"Signature-based IKE Phase 1 Main Mode" è simile al protocollo STS, ma due importanti differenze:

- Il protocollo STS lascia i certificati fuori dalla cifratura, mentre qui sono all'interno. La cifratura dei certificati permette l'anonimato.

- A differenza del protocollo STS, dove la firma non comprende la chiave di sessione condivisa, qui g^{xy} è contenuta nella firma. Come già in STS, la firma è esclusivamente verificata dalle parti che possiedono la chiave di sessione condivisa autentica.

Signature-based IKE Phase 1 Aggressive Mode

"Signature-based IKE Phase 1 Aggressive Mode" è una versione semplificata della modalità standard che non fa uso della cifratura ed è composta da tre scambi di messaggi invece di sei.
Gli scambi sono i seguenti:

1. $A \xrightarrow{HDR_A, SA_A, g^x, N_A, ID_A} B$
2. $A \xleftarrow{HDR_B, SA_B, g^y, N_B, ID_B, Cert_B, Sig_B} B$
3. $A \xrightarrow{HDR_B, Cert_A, Sig_A} B$

Questa modalità è molto simile al protocollo STS "Authentication-only" per via della mancanza della cifratura.
La differenza fondamentale è che qui la chiave di sessione condivisa g^{xy} è contenuta nella firma mentre nella versione semplificata del protocollo STS i messaggi firmati non comprendono la chiave di sessione.
In questo modo le firme sono verificabili esclusivamente da chi possiede la chiave di sessione, impedendo l'attacco "certificate-signature-replacement".
Tuttavia, questa modalità è vulnerabile al "denial of service attack" che rappresenta l'equivalente dell'attacco di Lowe sul protocollo STS.
In questo caso è A che crede di effettuare lo scambio con B, senza che B sia d'accordo.

4.5 Critiche a IPSec e IKE

I difetti di IPSec e IKE sono principalmente la complessità e la mancanza di chiarezza.
Questi sistemi contengono troppe opzioni e troppa flessibilità, infatti ci sono molti modi per raggiungere gli stessi obiettivi.
Oltre a ciò, hanno anche un'elevata complessità che potrebbe facilmente confondere i revisori esperti e impedire di vedere eventuali debolezze.

Inoltre possono indurre in errore gli implementatori, e causare la codifica di implementazioni difettose.
Alcune opzioni del sistema possono essere esse stesse pericolose per la sicurezza del protocollo, vediamo un esempio.
L'ESP contiene un campo opzionale, chiamato "campo di autenticazione ESP", che a sua volta include un valore di controllo di integrità (ICV) che è essenzialmente una firma digitale calcolata sull'ESP meno il campo di autenticazione in sè.
Questo varia a seconda della lunghezza dell'algoritmo di autenticazione utilizzato e in alcuni casi può anche essere omesso.
In questo esempio vediamo che è possibile omettere la protezione dell'integrità dei dati per un testo cifrato.
La cifratura senza integrità è pericolosa e generalmente gli algoritmi non possono fornire un'adeguata riservatezza senza una corretta protezione dell'integrità dei dati.
Altre critiche sono rivolte alla complessità di calcolo e comunicazione, inoltre i protocolli in IKE sono vulnerabili agli attacchi di "denial of service".
Nel 2002 W. Aiello, S.M. Bellovin, R. Canetti, J. Ioannidis, A.D. Keromytis e O. Reingold proposero un protocollo denominato "Just Fast Keying" (JFK) come successore di IKE per risolvere alcuni dei problemi.
JFK possiede le caratteristiche di sicurezza, semplicità, memory-DoS, computation-DoS, privacy, non negoziabilità, efficienza e perfect forward secrecy.

Bibliografia

[1] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, *Handbook of Applied Cryptography*

[2] Wenbo Mao, *Modern Cryptography: Theory and Practice*

[3] Laura Mei, *La PKI e l'attacco man in the middle*, tesi di laurea in algoritmi della teoria dei numeri e crittografia, III sessione a.a. 2014/2015, Università di Bologna

[4] M. Caiazza, G. Laccetti, G. Schmid, *Il metodo induttivo e la verifica di un protocollo crittografico*, Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni

[5] Luca Casoli, *Il problema del logaritmo discreto*, tesi di laurea in algoritmi della teoria dei numeri e crittografia, III sessione a.a. 2009/2010, Università di Bologna

[6] Open Systems Interconnection,
https://it.wikipedia.org/wiki/Open_Systems_Interconnection

[7] Integrated Services Digital Network,
https://it.wikipedia.org/wiki/Integrated_Services_Digital_Network

Ringraziamenti

Desidero innanzitutto ringraziare il professor Davide Aliffi per aver seguito pazientemente la mia tesi con estrema disponibilità e attenzione.
Ringrazio di cuore la mia famiglia che mi ha sostenuto con grande affetto durante questo percorso.
Un ringraziamento speciale va a Nicola che mi è sempre stato accanto con amore e pazienza supportandomi anche nei momenti peggiori, senza di lui tutto questo non sarebbe stato possibile.
Infine ringrazio tutti gli amici per aver contribuito ognuno a modo suo e per aver condiviso con me questi anni tra gioie e difficoltà.