Protocollo di autenticazione EKE: progetto di Network Security

Riccardo Orizio

A.A. 2015-2016

1 Requisiti del protocollo

L'idea dalla quale è nato il protocollo di autenticazione EKE è data dalla mancanza di qualsiasi tipo di autenticazione, singola e mutua, dell'algoritmo Diffie-Hellman, il quale scopo è la generazione di una chiave di sessione effimera.

Tale algoritmo consente a due entità di creare una chiave di sessione temporanea da zero, senza la necessità di conoscere alcun tipo di informazione aggiuntiva riguardante l'altro interlocutore. L'algoritmo è semplice, basato sulla trasmissione di tre pacchetti nei quali vengono scambiate le informazioni necessarie per la generazione della chiave da parte di entrambi i client, ovvero:

- **p**: un numero primo
- **g**: un generatore di Z_p^*
- $T_{a/b}$: chiave effimera parziale

Grazie al teorema di Eulero, entrambi i client sono in grado di derivare la stessa chiave effimera, seppur solamente parte della chiave viene scambiata. Il problema legato a questo algoritmo è la mancanza di autenticazione che lo rende soggetto al man-in-the-middle attack: Trudy può semplicemente rigirare le richieste ricevute da Alice a Bob ed utilizzare le sue risposte per creare una chiave effimera con Alice senza problemi. Per questi motivi EKE introduce l'autenticazione basata su una password conosciuta da entrambi i client e una coppia di challenge, i quali permettono di limitare il man-in-the-middle attack e garantire inoltre mutua autenticazione. La password condivisa dai client è importante per i primi messaggi, ma nel malaugurato caso in cui essa venisse recuperata da terzi, le uniche informazioni alla quali si avrebbero accesso sarebbero dei valori numerici di poca utilità.

Per poter eseguire il protocollo è quindi necessario conoscere esclusivamente una *password* condivisa tra i client che vogliono comunicare in una sessione protetta da una chiave effimera, o con un server[1], che tutti i client abbiano un nome univoco all'interno della rete nella quale il protocollo è attivo.

[1] In questo caso il server avrebbe tutte le password di tutti i client salvate in chiaro in un database

2 Il protocollo

Il protocollo si basa sullo scambio di quattro pacchetti totali nella quale viene generata la chiave effimera ed entrambi i client si autenticano a vicenda. La sequenza dei messaggi scambiati è la seguente, sapendo che entrambi conoscono la password e di conseguenza conoscono il valore $w = f(pwd)$:

1. **Client→Server**: [ClientID, A, g, p]

   ClientID: Identificativo del client
   A: $E_w(g^{S_a} \mod p) = E_w(T_a)$
   g: Generatore di Z_p^*
   p: Numero primo scelto casualmente dal client

2. **Server→Client**: [ServerID, B]

   ServerID: Identificativo del server (o secondo client)
   B: $E_w(g^{S_b} \mod p, c_1) = E_w(T_b, c_1)$

3. **Client→Server** $[E_k(c_1, c_2)]$

   k: chiave effimera $= T_b^{S_a} \mod p$

4. **Server→Client** $[E_k(c_2)]$

   k: chiave effimera $= T_a^{S_b} \mod p$

S_a ed S_b sono numeri casuali scelti nell'intervallo $[1, p)$, rispettivamente calcolati dal client e dal server; c_1 e c_2 sono numeri casuali usati come challenge per ottenere la mutua autenticazione tra i due interlocutori.

Errori di comunicazione possono sorgere quando i due client hanno una diversa password condivisa, di conseguenza non sono in grado di generare la stessa chiave effimera e le challenge non verranno verificate a causa della sbagliata decifrazione dei valori scambiati.

La funzione f è un *MDC* mentre E può essere un qualsiasi algoritmo di cifratura simmetrico o asimmetrico (ovviamente per la cifratura basata sulla chiave effimera generata, l'algoritmo scelto dovrà essere simmetrico). Nel progetto sono stati usati rispettivamente l'algoritmo *SHA-1* e *AES-128*.

3 Analisi

EKE garantisce una doppia autenticazione grazie alla doppia challenge scambiata tra i due utenti, i quali procedono con l'utilizzo della chiave effimera solamente se i valori decifrati ricevuti coincidono con quelli generati. Inoltre siamo protetti da eventuali dictionary attacks, indipendentemente dal fatto che siano on-line od off-line, in quanto le informazioni che si riescono a ricavare sono solamente dei numeri computazionalmente difficili da utilizzare per poter ricavare la chiave effimera.