Crittografia avanzata
Lezione del 14 Marzo 2011

Terminologia

- Modello standard
  - L'attaccante non è limitato se non dalla capacità computazionale e dal tempo disponibili
- Terze parti fidate (Trent)
  - Si assume che una certa entità svolga in modo assolutamente affidabile uno specifico compito
    - Verificare un'identità e firmare una coppia identità/chiave pubblica
    - Firmare una coppia {hash, orario}
    - ...

Oracolo

- Macchina che computa una funzione in modo “opaco” (black box)
  - All'oracolo si fanno domande e si ottengono risposte (esatte)
    - Alla stessa domanda si ottiene sempre la stessa risposta
  - Random oracle: ad ogni domanda risponde con una risposta casuale scelta uniformemente nel dominio di output
- Domande/risposte: numeri/stringhe di bit

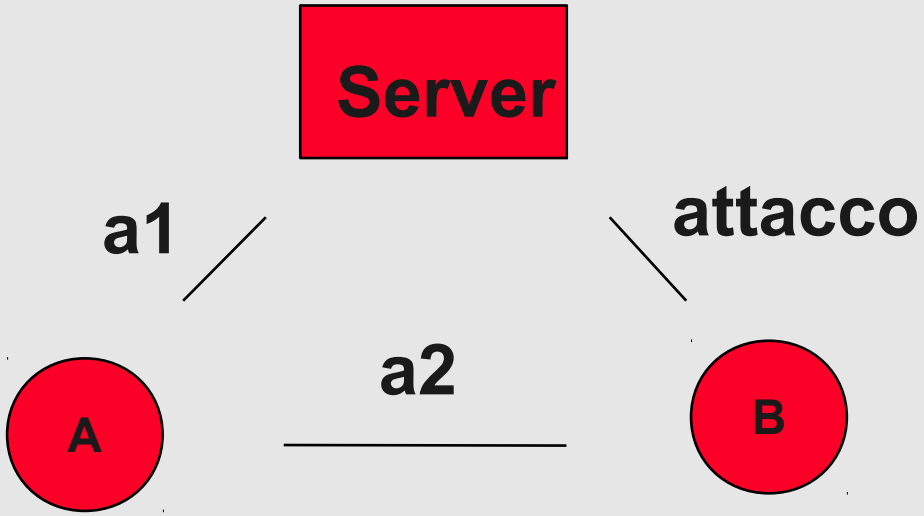
Crittografia e autenticazione

- Garantire l'autenticazione di messaggi o l'apertura di canali autenticati in presenza di Eve (eavesdropper) o Mallet (man-in-the-middle)
  - In assenza di questi problemi, da un punto di vista crittografico lo scambio di chiavi in chiaro andrebbe benissimo

Crittografia e protocolli (1)

- A dispone di una smart card (CIE?) con chiave privata
- Algoritmo 1 (RSA)
  - S sceglie un numero casuale N e lo invia ad A
  - A cifra N con la propria chiave privata M=S(N)
  - S riceve M e lo decifra con la chiave pubblica
- Algoritmo 2 (RSA)
  - B sceglie un numero casuale N’, lo cifra con al chiave pubblica di A M'=P(N') e lo invia ad A
  - A decifra M' con la propria chiave privata, e invia N' a B\
- Es. p=17,q=23,n=pq=391,(p-1)(q-1)=352,e=29,d=85

Crittografia e protocolli (2)



- A comunica con il server, autenticandosi con l'algoritmo 1
- A comunica con B, autenticandosi con l'algoritmo 2
- B non è in grado di vedere le comunicazioni fra A e il server
- B vuole impersonare A nelle comunicazioni con il server

Crittografia e protocolli (3)

- L’attacco:
  - A contatta B e chiede di autenticarsi
  - B contatta S affermando di essere A, e riceve N (10)
  - B invia N ad A, dicendo che è M'
  - A (de)cifra M’=N con la propria chiave privata, ottenendo M=S(N), e invia il risultato a B (198)
  - B invia M a S, e si autentica come A
  - Nota: l'attacco funziona anche con un algoritmo di crittografia perfetto

Considerazioni

- L'uso di crittografia non è una garanzia
- I singoli protocolli possono sembrare sicuri
  - I protocolli di autenticazione sono “scelti” da chi autentica, cioè S e B
- L'insicurezza nasce dall'operare con la propria chiave privata su un numero sul quale non si ha controllo
  - Per questo è importante non usare mai direttamente la chiave privata (es. cifrare un hash)

Come verificare la correttezza di un protocollo?

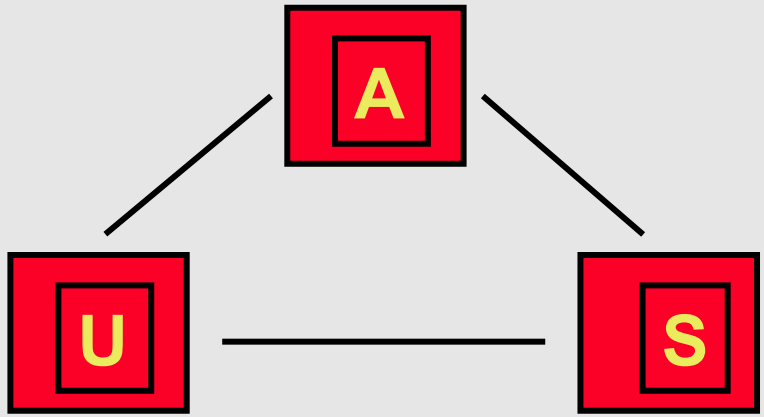
- C'è un notevole sviluppo di strumenti formali e tool di uso pratico
  - es. AVISPA
    - http://www.avispa-project.org/
- Si riescono a trovare vulnerabilità in protocolli reali
- I casi in cui la correttezza di un protocollo è dimostrata sono comunque pochi
  - Dipendono **molto** dalle ipotesi di partenza
  - Non è banale capire quanto sono realistiche

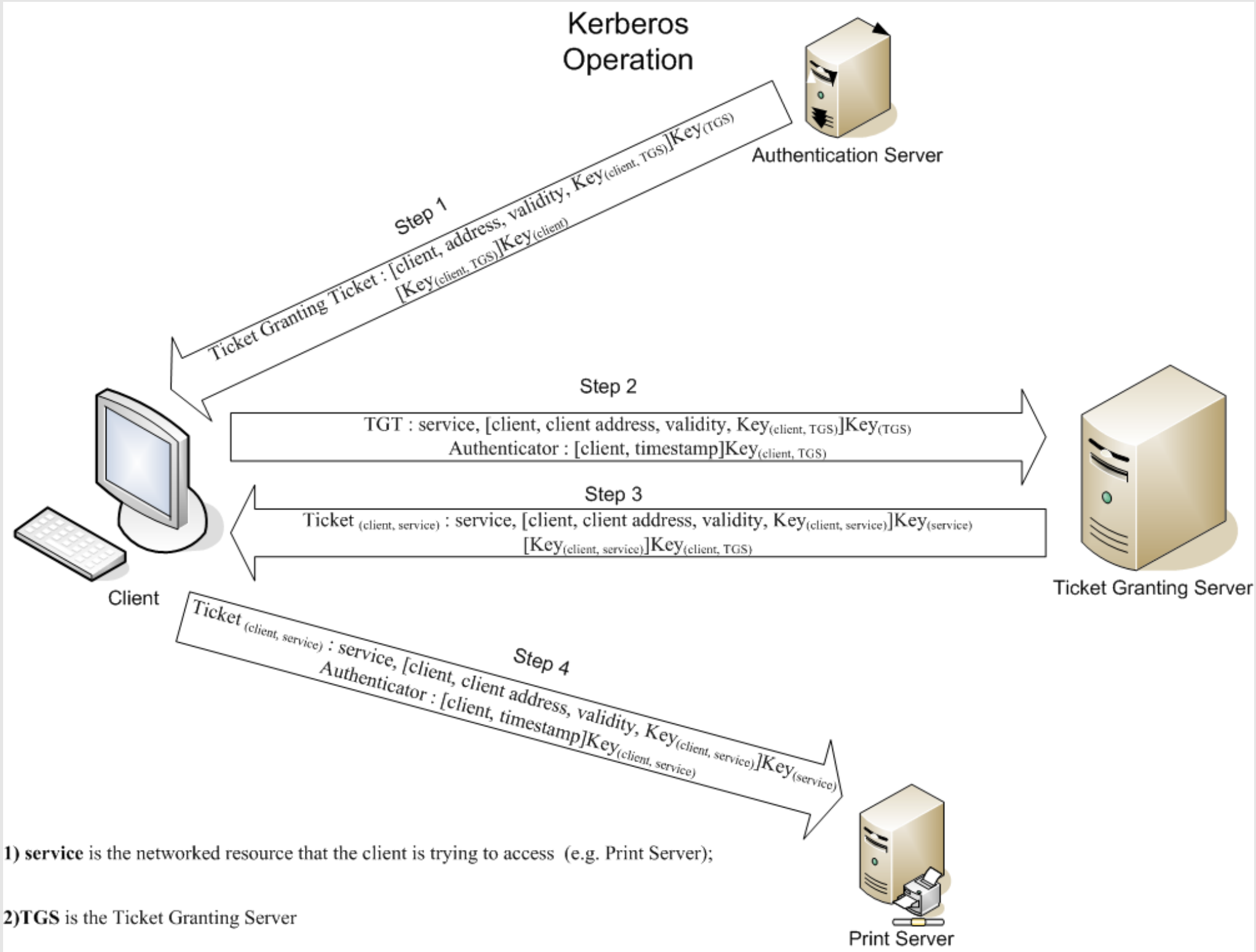
Authentication server

- Trusted third party
- Permette l'utilizzo di chiavi simmetriche
  - N chiavi per comunicare con N utenti
  - Chiavi effimere per la comunicazione fra utenti
- Problemi tipici:
  - Riuso delle chiavi

Kerberos: schema generale (1)

- Ogni utente ha una chiave segreta, ku
- Ogni servizio ha una chiave segreta, ks
- Entità coinvolte: l'utente U, il servizio S e l'Authentication Server A
- Schema Needham-Schröder simmetrico





1) service is the networked resource that the client is trying to access (e.g. Print Server);

2)TGS is the Ticket Granting Server

Kerberos: schema generale (2)

- Il server A conosce tutte le chiavi segrete
- Quando U vuole connettersi a S, chiede un ticket T ad A. Riceve $\{K, T\}_{k_u}$.
- T contiene:
  - l'indirizzo di U
  - l'ora
  - K, chiave per la sessione
  - il tutto cifrato con k_s

Kerberos: schema generale (3)

- U invia T a S
- S decifra il ticket con la propria chiave, e controlla che i dati contenuti siano corretti
- S risponde con l'ora + 1 crittografata con K
- Si ottiene così un'autenticazione reciproca

Kerberos: problemi

- L'authentication server conosce tutte le chiavi, e deve stare in linea
  - È adatto a contesti in cui c'è un unico gestore molto fidato, es. reti locali
    - È molto più critico di una CA
- Dipende dalla sincronizzazione degli orologi
  - C'è una finestra di vulnerabilità
    - È rilevante?
  - Replay cache (implementata?)

Realms

- Per migliorare la scalabilità, è possibile che utenti di un Realm vengano autenticati in un altro Realm se c'è una relazione di trust fra i TGS:
  - Essenzialmente, l'utente è autenticato localmente e ottiene un ticket per il TGS di un altro Realm

Vulnerabilità dell'implementazione

- Circa 60 vulnerabilità negli ultimi 4 anni
- Difetti software (overflow etc.)
  - Critici anche dato che il KDC è online
  - Anche in librerie (es. ASN.1)
- Pochi difetti nel protocollo in sé

Kerberos 4: known plaintext attack

- I ticket non hanno una checksum crittografica o altro
  - È possibile modificare il ticket senza che la manomissione venga rilevata se non per inconsistenze nei dati
  - Ticket splicing attack
    - » http://web.mit.edu/kerberos/advisories/MITKRB5-SA-2003-004-krb4

Scambio di chiavi mediante chiave pubblica: KDC (AC 3.1)

- Uso di un KDC:
  - A ottiene dal KDC la chiave pubblica di B
  - A usa la chiave pubblica di B per concordare una chiave effimera
  - Simile all'uso di CA, ma senza certificati (KDC in linea)

Scambio di chiavi mediante scambio diretto di chiave pubblica

- A invia a B la propria chiave pubblica
- B invia a d A la propria chiave pubblica
- Le chiavi pubbliche vengono utilizzate per concordare una chiave effimera
- Problema: Man in the Middle sullo scambio di chiavi pubbliche

Interlock protocol

- A e B si accordano su una chiave condivisa
- A invia a B “mezzo messaggio” Pa1 cifrato con la chiave condivisa
  - “mezzo” vuole dire chenon è sufficiente per decifrare Pa
- B invia ad A “mezza risposta” Pb1 cifrata
- A invia a B il resto del messaggio Pa2 cifrato
- B invia ad A il resto della risposta Pb2 cifrata

Interlock protocol

A B

Ea,b(Pa)<1>------->

<-------Ea,b(Pb)<1>

Ea,b(Pa)<2>------->

<-------Ea,b(Pb)<2>

Ea,b è la chiave condivisa (es. via Diffie Hellman)

Mallory?

- M sostituisce la propria chiave pubblica a quelle inviate da A e B, che quindi “cncordano” la chiave con lui
- M riceve Pa1, ma non è in grado di decifrarlo
- A non invia Pa2 finché non ha ottenuto Pb1
- A questo punto?

Tutto dipende da Pa e Pb

- Caso 1:
  - M può inventare un altro messaggio Pa' e mandare Pa'1 a B
  - B risponderà con Pb1; M inventa un nuovo messaggio Pb1' e invia Pb1'1 ad A
  - M riceve da A Pa2, decifra Pa. Invia Pa'2 a B, riceve Pb2 e decifra Pb. Invia Pb'2 ad A
  - Alla fine, A ha inviato Pa e B ha ricevuto Pa', B ha inviato Pb mentre A ha ricevuto Pb'. M conosce tutti e quattro i messaggi.
  - A e B si accorgono della differenza?

Conoscere i messaggi

A Z B

Ea,z(Pa)<1>------>

<------Ea,z(P?)<1>

Ea,z(Pa)<2>------>

Ez,b(Pa)<1>------>

<------Ez,b(Pb)<1>

Ez,b(Pa)<2>------>

<------Ez,b(Pb)<2>

Se l'ultimo passaggio non è completato, A non sa della compromissione (nessuno ha ricevuto messaggi sbagliati, ma un messaggio è andato perduto) e potrebbe ritentare la contrattazione. Però a questo punto M ha conosciuto Pa e PB, ed ha un Pb valido (se B non lo rinnova) da usare al posto di P?

Considerazioni

- Se lo scambio non ha successo, è necessario riprovare con Pa e Pb diversi
- Se non ha successo, non si “sa” che c'è Mallory, ma dopo un paio di tentativi si è comunque sicuri
  - Differenza fra teoria e pratica
- Come dividere Pa e Pb?
  - Non semplicemente a metà
  - Es. metà di ogni blocco cifrato (DES-CBC)

Concetto generale

- La logica di questo protocollo (dopo l'accordo sulla chiave, scambio di messaggi non concordati ma che le due parti “riconoscono”) è un concetto più generale, che ritroveremo (se riusciamo) nella cifratura VoiP

Cifrare per molti destinatari

- Problema: dati N su M destinatari, è oneroso inviare N copie del messaggio, ognuna cifrata con la chiave pubblica di un destinatario
- Soluzione semplice:
- Si cifra il messaggio M con una chiave simmetrica effimera K
- Si cifra K con ognuna delle N chiavi dei destinatari
- Continua ad essere oneroso per molti destinatari
  - Vedremo altri schemi

Secret splitting

- Voglio che un segreto S sia accessibile solo ad Alice e Bob quando sono insieme
- Esempio
  - Scelgo un numero casuale K1 lungo quanto S
  - Calcolo K2=K1 $\oplus$ S
  - Fornisco a Alice e Bob K1 e K2
  - S= K1 $\oplus$ K2
- Si può estendere facilmente a N persone, finché non ci sono tutte le chiavi in XOR non si ottiene S

Schemi a soglia

- Vogliamo che bastino *n* su *m* chiavi per decifrare il messaggio *T*
  - Problema delle chiavi e dei direttori: voglio che siano presenti almeno N direttori su M per aprire la cassaforte
- Schema di Shamir: per definire un polinomio di grado k servono k+1 punti.
- Definiamo un polinomio di grado *m-1*
  - $a_0+a_1X+a_2X^2+...+a_{m-1}X^{m-1}$
  - la cui costante a_0 è T e le altre sono casuali, e su questo scegliamo n punti:
- Servono m punti per definire il polinomio e quindi conoscere T

Il quitz della settimana

HIE 2	HE 1	DB
2	2	4
B 3	H 1	I
1	1	1
DGE 2	DE 1	DG

HI 2 D 1?

Soluzione

- Poche lettere (circa 10)
  - Saranno numeri?
- A quel punto è più facile associare numeri e simboli
  - 1 è =, ma gli altri?
  - HE 2 H = DE
    - 2 non può essere + o -
  - DGE 2 DE = DG
    - 2 non può essere +o x
    - 2 è ÷
  - ...

Il quitz della settimana

360	÷	30	=	12
÷		÷		+
2	×	3	=	6
=		=		=
180	÷	10	=	18