Protocolli di Sicurezza come Open System

Approcci alla Sicurezza (1)

- Molti protocolli oggi esistenti basano la propria affidabilità su opportune codifiche crittografiche
- La crittografia è uno strumento fondamentale, ma non sufficiente
- Purtroppo:
  - Molti lavori hanno identificato difetti in questi protocolli
  - Sono error-prone

Approcci alla Sicurezza (2)

- Altri approcci recenti si basano su
  - Esplorazione degli stati
  - Autenticazione logica
  - Algebre di processo
  - Tecniche di analisi statica
  - …

Open System (1)

- È possibile studiare I protocolli di sicurezza considerandoli come *open systems*
- Per open system si intende un sistema in cui una o più componenti non sono (completamente) specificate

Open System (2)

- La mancanza di completa specificazione può dipendere da
  - Mancanza di conoscenza di dettagli a livello implementativo
  - Incapacità di predire il comportamento di una componente all'interno del sistema
- In entrambi i casi si vuole garantire il corretto funzionamento del sistema, e il soddisfacimento di determinate proprietà

Open System (3)

- La verifica di un open system si basa sull'assunto seguente
  - Un open system soddisfa una determinata proprietà sse qualsiasi componente sia sostituita alla componente non specificata, l'intero sistema soddisfa quella proprietà
- Nelle algebre di processo (ad es., CSP o CCS) in cui il simbolo | rappresenta il parallelismo sono open systems:
  - A|(_)
  - A|B|(_)

Esempio

- Invio e ricezione di msg su una rete in cui "potrebbe" esserci un intruso
  - A vuole inviare un msg a B: A|B (si ignora l'intruso)
  - Meglio descrivere A|B|(_), dove la componente sconosciuta serve per considerare l'intruso

Analisi

- L'obiettivo dell'analisi è quello di garantire che un open system soddisfi proprietà di sicurezza indipendentemente dal comportamento di componenti maliziose (non specificate)
- Cioè, ∀ componente X vale S | X ⊨ p, dove
  - X è il possibile intruso
  - S è il sistema, costituito da partecipanti onesti
  - p è la proprietà di sicurezza
  - | è l'operatore di parallelismo
  - ⊨ è la relazione di verità

Commento (1)

- Lo statement principale dell'analisi assomiglia molto agli statement della logica della forma ∀ X vale X ⊨ p, in cui la formula p deve essere verificata per ogni struttura X
  - La differenza è che dobbiamo considerare X in combinazione con S
  - L'analisi si svolge riducendo il problema enunciato dallo statement principale a quello enunciato in generale dalla logica

Commento (2)

- L'analisi si svolge riducendo il problema enunciato dallo statement principale a quello enunciato in generale dalla logica

MessagePassing – Global Signature (1)

- È il modello che formalizza il formato dei messaggi scambiati tra gli agenti
  - In prima istanza consideriamo un modello semplificato, contenente un solo Client (C) e un solo EndServer (ES)
- La DASM relativa comprende una ASM per ciascuno dei 4 agenti del modello
  - Il KAS; il Client C; il TGS; l'EndServer ES
- Quindi, l'universo AGENT={KAS, C, TGS, ES}

MessagePassing – Global Signature (2)

- L'Universo MESSAGE contiene messaggi di due tipi, composti (concatenazione di dati) o crittografati (secondo una key), indicati rispettivamente con {X, X'} e $\{X, X'\}_{key}$
  - Per conoscere il tipo di un messaggio si usa la funzione type: MESSAGGE → {cleartext, encrypted}

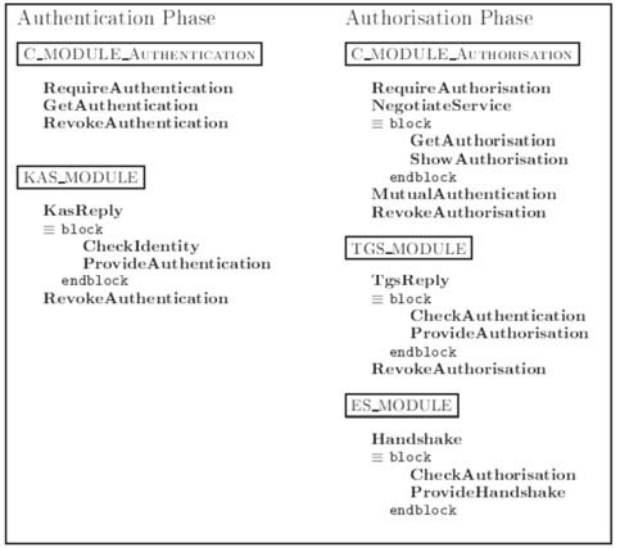
MessagePassing – Global Signature(3)

- Le chiavi, I ticket e le autenticazioni sono modellati da oggetti atomici
  - appartenenti rispettivamente agli universi KEY, TICKET e AUTH
- La funzione K: AGENT → KEY restituisce la chiave privata dell'agente, conosciuta solo dall'agente stesso e registrata nel db delle chiavi

MessagePassing – Global Signature (4)

- Le chiavi, I ticket e le autenticazioni sono spediti all'interno dei messaggi, da cui vengono estratti mediante le funzioni:
  - ExtractKey: MESSAGE → KEY
  - ExtractTicket: MESSAGE → TICKET
  - ExtractAuth: MESSAGE → AUTH
- Altre funzioni
  - sender, receiver: MESSAGE→ AGENT
  - mode: AGENT → {ReadyToSend, ReadyToReceive, ReadyToStart}

MessagePassing – Specifica



Bibliografia

- F. Martinelli, "Analysis of security protocols as open systems", *Theoretical Computer Science, 290*, pp.1057-1106, 2003