

## Verificarea protocoalelor de securitate

21 decembrie 2004

- modele ale protocoalelor și problemelor de securitate
- exemple tipice de protocoale și atacuri
- modelarea în logica BAN
- metode de verificare

Verificare formală. Curs 10

Marius Minea

## Protocoale de securitate - Importantă

Nevoia de comunicare secretă: încă din antichitate la fel și descoperirea cifrurilor, începutul criptografiei

Probleme de securitate: multiple aspecte: autentificare, integritate, confidențialitate, etc.

Soluțiile folosite sunt complexe și raționamentele asupra lor dificile. Securitatea unui protocol nu trebuie să se depindă de păstrarea secretă a algoritmului (NU: "security through obscurity")

- Erori subtile în protocoale existente au fost descoperite uneori după foarte mult timp (17 ani, într-un caz)

⇒ riscuri mari în cazul compromiterii unui algoritm slab, secret și deci neanalizat de specialiști

⇒ importanța verificării formale cu atât mai mare

Verificare formală. Curs 10

Marius Minea

Verificarea protocoalelor de securitate

3

### Chei comune și chei publice

O problemă fundamentală: stabilirea unui canal sigur (criptat) de comunicație între două entități

#### Criptare cu cheie comună

- cheie comună cunoscută doar de cei doi participanți
- cheia de decriptare e obținută simplu din cea de criptare (conventional se consideră aceeași)
- exemplu: Data Encryption Standard (1975)

#### Criptare cu chei publice

- fiecare participant  $A$ : pereche de chei
- publică,  $K_a$ : criptare / secretă,  $K_a^{-1}$ : decriptare.
- $A$  trimite  $K_b(K_a^{-1}(M))$ : decriptat doar de  $B$ , sigur de la  $A$
- ex. Diffie-Hellman, Rivest-Shamir-Adleman (1976)

Verificare formală. Curs 10

Marius Minea

Verificarea protocoalelor de securitate

4

### Modele pentru protocoale

[Dolev & Yao '83]: importanța unor prezumții clare în modelarea, analiza și verificarea protocoalelor:

- Într-un sistem cu chei publice:
  - funcțiile de criptare nu se pot sparge
  - directorul de chei publice e permanent integru
  - fiecare are acces la toate cheile publice  $E_X, \forall X$
  - numai  $X$  are acces la cheia de decriptare  $D_X$
- Un protocol între doi participanți nu necesită asistența unui al treilea pentru criptare sau decriptare
- Într-un protocol uniform, toate perechile comunicante folosesc același format pentru mesaje

Verificare formală. Curs 10

Marius Minea

Verificarea protocoalelor de securitate

5

### Exemple de atacuri

Protocolul 1:

- $A \rightarrow B: E_B(M)$
- $B \rightarrow A: E_A(M)$

Atacul 1:

- $A \rightarrow B: E_B(M)$ , interceptat de  $Z$
- $Z \rightarrow B: E_B(M)$
- $B \rightarrow Z: E_Z(M)$ ;  $Z$  decodifică  $M$

Protocolul 2:

- $A \rightarrow B: E_B(E_B(M)A)$
- $B \rightarrow A: E_A(E_A(M)B)$

Atacul 2:

- $Z \rightarrow A: E_A(E_A(E_A(M)B)Z)$
- $A \rightarrow Z: E_Z(E_Z(E_A(M)B)A)$
- $Z$  decodifică  $E_A(M)B$ , deci are  $E_A(M)$
- $Z \rightarrow A: E_A(E_A(M)Z)$
- $A \rightarrow Z: E_Z(E_Z(M)A)$ , deci  $Z$  are  $M$

Verificare formală. Curs 10

Marius Minea

Verificarea protocoalelor de securitate

6

### Dolev & Yao: Modelul atacatorului (intrusului)

Sabotorii sunt "activi": ei pot asculta linia pentru a capta mesaje, și apoi fac tot posibilul pentru a le descifra:

- poate obține orice mesaj din rețea
- e un utilizator legitim al rețelei, în particular poate iniția o conversație cu orice utilizator
- va avea ocazia să recepționeze mesaje de la orice utilizator  $A$  (mai general, se presupune că orice utilizator  $B$  poate deveni receptor pentru orice utilizator  $A$ )

Verificare formală. Curs 10

Marius Minea

## Primele rezultate formale

Dolev & Yao discută două tipuri de protocoale, definite prin operațiile permise în fiecare din ele:

1. Protocoale în cascadă
  - criptare cu orice cheie publică
  - decriptare cu cheia proprie
2. Protocoale cu etichetare cu nume (name stamp). Suplimentar:
  - adăugarea la mesaj a unui nume de participant
  - ștergerea numelui unui anumit participant
  - ștergerea oricărui nume

Problema corectitudinii devine o problemă de rescriere pentru șiruri dintr-un alfabet, decidabilă în timp polinomial  
Dar nedecidabilă pentru clase mai complexe

Verificare formală. Curs 10

Marius Minea

## Protocoale de autentificare

protocoale prin care participanții se conving reciproc de identitatea lor și fie stabilesc (chei) secrete comune pentru comunicare, fie recunosc utilizarea cheilor secrete ale partenerilor.

– cele mai studiate protocoale de securitate din literatură

Notatii:  $A, B$ : participanți.  $S$ : server de autentificare.  $N_a, N_b$ : "nonce" (de la: number once) = număr aleator generat pentru a evita reutilizarea mesajelor vechi de către un intrus  
 $\{X\}_K$ : mesajul  $X$  criptat cu cheia  $K$

[Needham & Schroeder '78] "Using Encryption for Authentication in Large Networks of Computers": articol clasic. Printre altele, sunt primii care au prezis importanța metodelor formale de verificare.

Verificare formală. Curs 10

Marius Minea

## Protocolul Needham-Schroeder cu cheie comună

- (1)  $A \rightarrow S: A, B, N_a$   
 $A$  anunță serverului  $S$  intenția de a comunica cu  $B$   
(și garantează prospețimea mesajului cu un "nonce"  $N_a$ )
- (2)  $S \rightarrow A: \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$   
 $S$  transmite lui  $A$  cheia  $K_{ab}$ , împreună cu un mesaj criptat pentru  $B$ , pe care  $A$  îl retransmite acestuia:
- (3)  $A \rightarrow B: \{K_{ab}, A\}_{K_{bs}}$   
 $B$  extrage cheia  $K_{ab}$  și anunță pe  $A$  prin transmiterea unui nonce  $N_b$ :
- (4)  $B \rightarrow A: \{N_b\}_{K_{ab}}$   
 $A$  confirmă retransmițând un mesaj bazat pe  $N_b$   
(în mod convențional, decrementat cu 1):
- (5)  $A \rightarrow B: \{N_b - 1\}_{K_{ab}}$   
Acum, ambii participanți știu că pot comunica cu  $K_{ab}$

Verificare formală. Curs 10

Marius Minea

## Needham-Schroeder: vulnerabilitate (1)

[Denning & Sacco, 1981]

Problema: un intrus care a urmărit o sesiune anterioară poate să forțeze pe  $B$  să accepte o cheie veche, potențial compromisă

Intrusul  $I$  impersonază pe  $A$  (notăm  $I(A)$ ) și trimite lui  $B$  mesajul (3) din sesiunea anterioară, cu cheia veche  $K_c$ :

- (3)  $I(A) \rightarrow B: \{K_c, A\}_{K_{bs}}$
- (4)  $B \rightarrow I(A): \{N_b\}_{K_c}$
- (5)  $I(A) \rightarrow B: \{N_b - 1\}_{K_c}$

Pericolul:  $I$  are timp practic nelimitat să compromită cheia  $K_c$

Corecție: etichete de timp (timestamps) sau nonce suplimentar

Verificare formală. Curs 10

Marius Minea

## Needham-Schroeder: vulnerabilitate (2)

- [Lowe '95] găsește eroare în protocolul cu chei publice (după 17 ani!)
- (1)  $A \rightarrow B: A, B, \{N_a, A\}_{K_b}$   $A$  cere comunicarea, transmite nonce  $N_a$
  - (2)  $B \rightarrow A: B, A, \{N_a, N_b\}_{K_a}$   $B$  răspunde cu nonce propriu  $N_b$
  - (3)  $A \rightarrow B: A, B, \{N_b\}_{K_b}$   $A$  confirmă recepția

Atac cu două sesiuni concurente:  $A$  inițiază sesiunea  $\alpha$  cu intrusul  $I$ ; acesta impersonază pe  $A$  în sesiunea  $\beta$  cu  $B$

- ( $\alpha.1$ )  $A \rightarrow I: A, I, \{N_a, A\}_{K_i}$
- ( $\beta.1$ )  $I(A) \rightarrow B: A, B, \{N_a, A\}_{K_b}$
- ( $\beta.2$ )  $B \rightarrow I(A): B, A, \{N_a, N_b\}_{K_a}$
- ( $\alpha.2$ )  $I \rightarrow A: I, A, \{N_a, N_b\}_{K_a}$
- ( $\alpha.3$ )  $A \rightarrow I: A, I, \{N_b\}_{K_i}$
- ( $\beta.3$ )  $I(A) \rightarrow B: A, B, \{N_b\}_{K_b}$

Descoperit: cu un model-checker (FDR) pentru limbajul CSP  
Corecție: includerea numelui emițătorului, criptat, în mesajul (2)

Verificare formală. Curs 10

Marius Minea

## Tipuri de atacuri

"Cryptography is not broken, it is circumvented" – A. Shamir

Clark & Jacob, "A Survey of Authentication Protocol Literature, '97:

- Atacuri de "prospețime" (freshness attacks)
  - un mesaj (sau fragment) dintr-o sesiune de comunicație anterioară este memorat și inserat de un intrus într-o nouă sesiune
- Atacuri cu erori de tip (type flaws)
  - Un mesaj de compus din câmpuri, fiecare cu o anumită interpretare (date, număr unicat, numele unui participant, valoarea unei chei)
  - Atacul e bazat pe acceptarea unui mesaj cu o altă interpretare (pe câmpuri de biți) decât cea cu care a fost transmis inițial.

Verificare formală. Curs 10

Marius Minea

### Tipuri de atacuri (cont.)

- Atacuri în sesiuni paralele
  - două sau mai multe sesiuni concurente ale aceluiași protocol
  - mesajele dintr-o sesiune folosite în atacul alteia
- Atacuri dependente de implementare
  - atacurile de tip pot fi eliminate dacă reprezentarea componentelor mesajului conține redundanță pt. a distinge tipul
  - interacțiunea dintre protocol și metoda de criptare (ex. schimbarea unui bit în criptarea bit cu bit)
- Atacuri la integritatea cheii (binding attacks)
  - inducerea în eroare asupra cheii publice a partenerului (înlocuirea cu cheia publică a intrusului)
- ... și multe altele

### Modelarea în logica BAN

[Burrows, Abadi, Needham '89: "A logic of authentication"]

- cea mai importantă metodă de modelare în logică
- o logică despre *convîngeri* (logic of belief), spre deosebire de logicile despre *cunoaștere* (logic of knowledge)
- despre ceea ce *crede* fiecare participant că e adevărat

Scopul: de a exprima cu precizie:

- prezumțiile inițiale despre funcționarea unui protocol
- convingerile finale la care ajung participanții

Exemple:

- ce realizează (atinge) protocolul ?
- necesită mai multe prezumții decât alt protocol ?
- transmite/criptează ceva care nu e necesar ?

### Logica BAN: noțiuni de bază

$P \models X$	$P$ crede $X$
$P \triangleleft X$	$P$ vede (primește, citește) mesajul $X$
$P \rightsquigarrow X$	$P$ a spus (transmis) $X$ cândva în trecut
$P \Rightarrow X$	$P$ are jurisdicție asupra lui $X$ . $P$ este o autoritate în materie de $X$ (ex. o cheie) și trebuie crezut
$\sharp(X)$	$X$ este proaspăt (nu a fost trimis până acum)
$P \stackrel{K}{\leftrightarrow} Q$	$P$ și $Q$ pot folosi cheia comună $K$ pentru a comunica
$\overset{K}{\rightarrow} P$	$P$ are cheia publică $K$
$P \overset{X}{\leftarrow} Q$	$X$ este un secret cunoscut doar de $P$ și $Q$
$\{X\}_K$	mesajul $X$ criptat cu cheia $K$
$\langle X \rangle_Y$	$X$ combinat cu secretul $Y$ (pentru identificare)

### Logica BAN: reguli de inferență

Reguli cu privire la semnificația mesajelor:

- pentru chei comune:

$$\frac{P \models Q \stackrel{K}{\leftrightarrow} P, P \triangleleft \{X\}_K}{P \models Q \rightsquigarrow X}$$

- pentru chei publice:

$$\frac{P \models \overset{K}{\rightarrow} Q, P \triangleleft \{X\}_{K^{-1}}}{P \models Q \rightsquigarrow X}$$

- pentru secrete comune

$$\frac{P \models Q \overset{X}{\leftarrow} P, P \triangleleft \langle X \rangle_Y}{P \models Q \rightsquigarrow X}$$

Reguli referitoare la mesaje recente:

$$\frac{P \models \sharp(X), P \models Q \rightsquigarrow X}{P \models Q \models X} \quad \frac{P \models \sharp(X)}{P \models \sharp(X, Y)}$$

### Logica BAN: reguli de inferență (cont.)

Regula de jurisdicție:

$$\frac{P \models Q \Rightarrow X, P \models Q \models X}{P \models X}$$

Compoziție:  $P$  crede un ansamblu  $\Leftrightarrow$  crede părțile

Proiecție:  $P$  a spus un ansamblu  $\Rightarrow$  a spus părțile

$$\frac{P \models X, P \models Y}{P \models (X, Y)} \quad \frac{P \models (X, Y)}{P \models X} \quad \frac{P \models Q \rightsquigarrow (X, Y)}{P \models Q \rightsquigarrow X}$$

Reguli de decriptare, de ex.

$$\frac{P \models \overset{K}{\rightarrow} Q, P \triangleleft \{X\}_{K^{-1}}}{P \triangleleft X}$$

Bidirecționalitatea cheilor și secretelor între participanți:

$$\frac{P \models R \overset{K}{\leftrightarrow} R'}{P \models R' \overset{K}{\leftrightarrow} R}$$

### Exemplu: Needham-Schroeder cu chei comune

- (1)  $A \rightarrow S: A, B, N_a$
- (2)  $S \rightarrow A: \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
- (3)  $A \rightarrow B: \{K_{ab}, A\}_{K_{bs}}$
- (4)  $B \rightarrow A: \{N_b\}_{K_{ab}}$
- (5)  $A \rightarrow B: \{N_b - 1\}_{K_{ab}}$

Idealizăm protocolul: în loc de mesaje de biți, transmitem *formule logice*, corespunzătoare semnificației mesajelor:

- (1) Mesajul 1 e doar o solicitare, nu are valoare logică
- (2)  $S \rightarrow A: \{N_a, B, (A \overset{K_{ab}}{\leftrightarrow} B), \sharp(A \overset{K_{ab}}{\leftrightarrow} B), \{A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}\}_{K_{as}}$
- (3)  $A \rightarrow B: \{A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}$
- (4)  $B \rightarrow A: \{N_b, (A \overset{K_{ab}}{\leftrightarrow} B)\}_{K_{ab}}$  de la  $B$
- (5)  $A \rightarrow B: \{N_b, (A \overset{K_{ab}}{\leftrightarrow} B)\}_{K_{ab}}$  de la  $A$

Pornim de la premisele (abreviem  $P, Q \models X$  pt.  $P \models X$  și  $Q \models X$ ):

$$A, S \models A \xleftrightarrow{K_{as}} S \quad B, S \models B \xleftrightarrow{K_{bs}} S \quad S \models A \xleftrightarrow{K_{sb}} B$$

$$A, B \models (S \models A \xleftrightarrow{K} B) \quad A \models (S \models \#(A \xleftrightarrow{K} B))$$

(o bună cheie e "proaspătă", premisă explicitată aici)

$$A \models \#(N_a) \quad B \models \#(N_b) \quad S \models \#(A \xleftrightarrow{K_{sb}} B)$$

$$\text{Din } A \models \#(N_a), \text{ și (2) deducem: } A \models S \models A \xleftrightarrow{K_{sb}} B, A \models S \models \#(A \xleftrightarrow{K_{sb}} B)$$

$$\text{iar apoi din regula de jurisdicție: } A \models A \xleftrightarrow{K_{sb}} B \quad A \models \#(A \xleftrightarrow{K_{sb}} B)$$

$$\text{După primirea fragmentului (3) de la } A, \text{ deducem: } B \models S \vdash A \xleftrightarrow{K_{sb}} B$$

$$\text{Nu se poate obține } B \models A \xleftrightarrow{K_{sb}} B \text{ fără premisa } B \models \#(A \xleftrightarrow{K} B) \text{ (!!)} \text{!}$$

Din prospețimea mesajelor (4) și (5) deducem  $A \models B \models A \xleftrightarrow{K_{sb}} B$  și  $B \models A \models A \xleftrightarrow{K_{sb}} B$ , deci fiecare participant e convins atât de validitatea cheii, cât și că acest lucru e crezut și de celălalt.

Răționamentul evidențiază premisa care din cele văzute e periculoasă, permițând unui intrus să substituie o cheie compromisă.

### Logica BAN: aplicabilitate și limitări

- permite demonstrarea unor proprietăți despre protocoale
- dacă nu se poate demonstra: motive serioase de dubiu
- poate identifica premise dubioase, neexplicitate altfel

Dar:

- logică monotonă: un fapt existent nu poate fi retractat
- nu tratează notiunea de confidențialitate a cheilor sau compromiterea acesteia (de exemplu: transmiterea unei chei în clar)

### Verificare: model checking

Se poate aplica un model checker generic, modelând intrusul și acțiunile sale posibile.

Sau: model checker specializat, ex. BRUTUS [Clarke, Marrero, Jha]  
Modelarea implicită a intrusului: o relație  $\vdash$  prin care intrusul poate deriva mesaje  $m$  dintr-un set inițial de informații  $I$ :

- dacă  $m \in I$  atunci  $I \vdash m$
- concatenare: dacă  $I \vdash m_1$  și  $I \vdash m_2$  atunci  $I \vdash m_1 \cdot m_2$
- proiecție: dacă  $I \vdash m_1 \cdot m_2$  atunci  $I \vdash m_1$  și  $I \vdash m_2$
- criptare: dacă  $I \vdash m$  și  $I \vdash k$  atunci  $I \vdash \{m\}_k$
- decriptare: dacă  $I \vdash \{m\}_k$  și  $I \vdash k^{-1}$  atunci  $I \vdash m$

Protocolul: compoziție asincronă între participanți și intrus.

Intrusul poate asculta orice, și poate șterge, modifica sau adăuga mesaje cf. setului său de informații

### Verificare: generare de teorii

Exemplu: RVChecker, REVERE [Kindred & Wing]

Utilizat pentru o logică de tipul BAN extinsă

Generarea de teorii (pentru anumite logici simple):

- o metodă sintactică de demonstrare de teoreme bazată pe saturare
- produce o reprezentare finită a unei teorii posibil infinite (toate teoremele generate din niște ipoteze și reguli)
- terminarea bazată pe limitarea aplicării regulilor de inferență care pot genera concluzii de dimensiuni mai mari decât premisele

Combinarea cu model checking:

- o premisă dubioasă găsită prin generarea de teorii: folosită pentru modelarea unui atac
- reciproc, un contraxemplu, modelat sub formă de deducție logică poate identifica o premisă dubioasă

### Verificare: demonstrare de teoreme

Principalul dezavantaj pt. model checking: e necesar un model finit, deci limitarea a numărului de participanți și sesiuni

Demonstratoarele de teoreme nu au această limitare

Răționament/rescriere în Prolog: Interrogator [Millen'87]

NRL Protocol Analyzer [Meadows et al.]

- combinație theorem-proving + model checking
- pornește de la o stare de eroare (dorit inaccesibilă)
- caută înapoi folosind tehnici inductive

Athena [Song et al., CMU/Berkeley]

- reduce evaluarea unei formule la explorarea unui spațiu finit
- reprezentare (strand space) bazată pe cauzalitate și nu pe execuții individuale  $\Rightarrow$  reduce mult spațiul stărilor
- explorarea de stări parametrizate cu variabile libere