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

Protocoale de securitate - Importanţă

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

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

Soluţiile folosite sunt complexe şi raţionamentele 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 şi deci neanalizat de specialişti
⇒ importanţa verificării formale cu atât mai mare

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 (convențional 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)

Modele pentru protocoale

[Dolev & Yao '83]: importanţa unor prezumţii clare în modelarea, analiza şi verificarea protocoalelor:

1) Într-un sistem cu chei publice:
a) funcţiile de criptare nu se pot sparge
b) directorul de chei publice e permanent integru
c) fiecare are acces la toate cheile publice E_X, $\forall X$
d) numai X are acces la cheia de decriptare D_X

2) Un protocol între doi participanţi nu necesită asistenţa unui al treilea pentru criptare sau decriptare

3) Într-un protocol uniform, toate perechile comunicante folosesc acelaşi format pentru mesaje

Exemple de atacuri

Protocolul 1:

(1) $A \to B$: $E_B(M)$

(2) $B \to A$: $E_A(M)$

Protocolul 2:

(1) $A \to B$: $E_B(E_B(M)A)$

(2) $B \to A$: $E_A(E_A(M)B)$

Atacul 1:

(1) $A \to B$: $E_B(M)$, interceptat de Z

(2) $Z \to B$: $E_B(M)$

(3) $B \to Z$: $E_Z(M)$; Z decodifică M

Atacul 2:

(1) $Z \to A$: $E_A(E_A(E_A(M)B)Z)$

(2) $A \to Z$: $E_Z(E_Z(E_A(M)B)A)$

(3) Z decodifică $E_A(M)B$, deci are $E_A(M)$

(4) $Z \to A$: $E_A(E_A(M)Z)$

(4) $A \to Z$: $E_Z(E_Z(M)A)$, deci Z are M

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:
a) poate obține orice mesaj din rețea
b) e un utilizator legitim al rețelei, în particular poate iniția o conversație cu orice utilizator
c) 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)

Primele rezultate formale

Dolev & Yao discută două tipuri de protocoale, definite prin operaţiile 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
– ştergerea numelui unui anumit participant
– ştergerea oricărui nume

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

Protocoale de autentificare

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

– cele mai studiate protocoale de securitate din literatură

Notaţii: A, B: participanţi. 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 importanţa metodelor formale de verificare.

Protocolul Needham-Schroeder cu cheie comună

(1) $A \rightarrow S$: A, B, N_a

A anunţă serverului S intenţia de a comunica cu B

(şi garantează prospeţimea 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} şi anunţă pe A prin transmiterea unui nonce N_b:

(4) $B \rightarrow A$: $\{N_b\}_{K_{ab}}$

A confirmă retransmiţând un mesaj bazat pe N_b

(în mod convenţional, decrementat cu 1):

(5) $A \rightarrow B$: $\{N_b - 1\}_{K_{ab}}$

Acum, ambii participanţi ştiu că pot comunica cu K_{ab}

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 impersonează 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

Needham-Schroeder: vulnerabilitate (2)

[Lowe '95] găsește eroare în protocolul cu chei publice (după 17 ani!)

(1) $A \to B$: $A, B, \{N_a, A\}_{K_b}$ A cere comunicarea, transmite nonce N_a

(2) $B \to A$: $B, A, \{N_a, N_b\}_{K_a}$ B răspunde cu nonce propriu N_b

(3) $A \to B$: $A, B, \{N_b\}_{K_b}$ A confirmă recepția

Atac cu două sesiuni concurente: A inițiază sesiunea α cu intrusul I; acesta impersonează pe A în sesiunea β cu B

(α.1) $A \to I$: $A, I, \{N_a, A\}_{K_i}$

(β.1) $I(A) \to B$: $A, B, \{N_a, A\}_{K_b}$

(β.2) $B \to I(A)$: $B, A, \{N_a, N_b\}_{K_a}$

(α.2) $I \to A$: $I, A, \{N_a, N_b\}_{K_a}$

(α.3) $A \to I$: $A, I, \{N_b\}_{K_i}$

(β.3) $I(A) \to B$: $A, B, \{N_b\}_{K_b}$

Descoperit: cu un model-checker (FDR) pentru limbajul CSP

Corecție: includerea numelui emițătorului, criptat, în mesajul (2)

Tipuri de atacuri

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

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

• Atacuri de "prospeţime" (freshness attacks)

– un mesaj (sau fragment) dintr-o sesiune de comunicaţie anterioară este memorat şi 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 biţi) decât cea cu care a fost transmis iniţial.

Tipuri de atacuri (cont.)

- Atacuri în sesiuni paralele
– două sau mai multe sesiuni concurente ale aceluiaşi protocol
– mesajele dintr-o sesiune folosite în atacul alteia

- Atacuri dependente de implementare
– atacurile de tip pot fi eliminate dacă reprezentarea componentelor mesajului conţine redundanţă pt. a distinge tipul
– interacţiunea dintre protocol şi 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)

- ... şi 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 *convingeri* (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:
– prezumţiile iniţiale despre funcţionarea unui protocol
– convingerile finale la care ajung participanţii

Exemple:
– ce realizează (atinge) protocolul ?
– necesită mai multe prezumţii decât alt protocol ?
– transmite/criptează ceva care nu e necesar ?

Logica BAN: noţiuni de bază

$P \models\!\equiv X$	P crede X
$P \lhd X$	P vede (primeşte, citeşte) mesajul X
$P \mid\!\sim X$	P a spus (transmis) X cândva în trecut
$P \models\!\Rightarrow X$	P are jurisdicţie asupra lui X. P este o autoritate în materie de X (ex. o cheie) şi trebuie crezut
$\sharp(X)$	X este proaspăt (nu a fost trimis până acum)
$P \overset{K}{\leftrightarrow} Q$	P şi Q pot folosi cheia comună K pentru a comunica
$\overset{K}{\mapsto} P$	P are cheia publică K
$P \overset{X}{=\!=} Q$	X este un secret cunoscut doar de P şi Q
$\{X\}_K$	mesajul X criptat cu cheia K
$\langle X \rangle_Y$	X combinat cu secretul Y (pentru identificare)

Logica BAN: reguli de inferenţă

Reguli cu privire la semnificaţia mesajelor:

– pentru chei comune:

$$\frac{P \models Q \overset{K}{\leftrightarrow} P,\ P \lhd \{X\}_K}{P \models Q \mid\sim X}$$

– pentru chei publice:

$$\frac{P \models \overset{K}{\mapsto} Q,\ P \lhd \{X\}_{K^{-1}}}{P \models Q \mid\sim X}$$

– pentru secrete comune

$$\frac{P \models Q \overset{Y}{\rightleftharpoons} P,\ P \lhd \langle X \rangle_Y}{P \models Q \mid\sim X}$$

Reguli referitoare la mesaje recente:

$$\frac{P \models \sharp(X),\ P \models Q \mid\sim X}{P \models Q \models X} \qquad \frac{P \models \sharp(X)}{P \models \sharp(X, Y)}$$

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

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

Reguli de decriptare, de ex.

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

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

Idealizăm protocolul: în loc de mesaje de biţi, transmitem *formule logice*, corespunzătoare semnificaţiei mesajelor:

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

Raționament: Needham-Schroeder cu chei comune

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

$A, S \models A \overset{K_{as}}{\leftrightarrow} S \qquad B, S \models B \overset{K_{bs}}{\leftrightarrow} S \qquad S \models A \overset{K_{ab}}{\leftrightarrow} B$

$A, B \models (S \models\!\!> A \overset{K}{\leftrightarrow} B) \qquad A \models (S \models\!\!> \sharp(A \overset{K}{\leftrightarrow} B))$

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

$A \models \sharp(N_a) \qquad B \models \sharp(N_b) \qquad S \models \sharp(A \overset{K_{ab}}{\leftrightarrow} B)$

Din $A \models \sharp(N_a)$, și (2) deducem: $A \models S \models A \overset{K_{ab}}{\leftrightarrow} B$, $A \models S \models \sharp(A \overset{K_{ab}}{\leftrightarrow} B)$

iar apoi din regula de jurisdicție: $A \models A \overset{K_{ab}}{\leftrightarrow} B \qquad A \models \sharp(A \overset{K_{ab}}{\leftrightarrow} B)$

După primirea fragmentului (3) de la A, deducem: $B \models S \mid\!\sim A \overset{K_{ab}}{\leftrightarrow} B$

Nu se poate obține $B \models A \overset{K_{ab}}{\leftrightarrow} B$ fără premisa $B \models \sharp(A \overset{K}{\leftrightarrow} B)$ (!!)

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

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

Logica BAN: aplicabilitate și limitări

– permite demonstrarea unor proprietăţi 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 noţiunea de confidenţialitate 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 deducţie 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 participanţi şi sesiuni
Demonstratoarele de teoreme nu au această limitare

Raţionament/rescriere în Prolog: Interrogator [Millen'87]
NRL Protocol Analyzer [Meadows et al.]
– combinaţie theorem-proving + model checking
– porneşte 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 spaţiu finit
– reprezentare (strand space) bazată pe cauzalitate şi nu pe execuţii individuale $\Rightarrow$ reduce mult spaţiul stărilor
– explorarea de stări parametrizate cu variabile libere