14eme Rencontre Professeurs et Enseignants-Chercheurs
Vérification de protocoles cryptographiques

Denis Lugiez
Laboratoire d'Informatique Fondamentale

Protocole cryptographique

programme qui permet d'établir une interaction entre des participants dans un environnement non sûr en garantissant certaines propriétés.

- E-commerce (paiement, enchères, non-répudiation,...)
- E-administration (impots, dossier médical, vote électronique,...)
- Réseaux de machines mobiles distantes (portables, téléphones,...)

Utilisation de la cryptographie (AES, RSA,..) pour chiffrer les données sensibles et assurer les propriétés voulues.

La question

A quoi bon un coffre-fort indestructible
si la combinaison est récupérable sans effort...

La question

E-voting :

L'autorité attribue à chaque votant une clé publique K_{pub} pour chiffrer son vote (publication à la mairie). Seule l'autorité possède K_{pub}^{-1} et peut déchiffrer un message chiffré avec K_{pub}.

- Vote entre Hollande et Sarkozy : chiffrer le nom du candidat avec la clef publique et l'envoyer à l'autorité qui est la seule à pouvoir déchiffrer.
- Propriété requise : confidentialité du vote.
- Faille logique : on peut intercepter mon vote et trouver mon choix sans connaitre K_{pub}^{-1} !

L'attaque repose sur une faille logique du protocole.
Vérification des protocoles :

- donner un modèle formel permettant de trouver les failles en applicant un algorithme de vérification.
- ne vérifie pas l'implémentation du protocole (i.e. le programme C qui tourne effectivement).
- autre approche : modèle computationel. Analyse plus proche de la théorie de l'information mais relation forte avec le modèle formel.

Protocole d'authentification (Needham-Schroeder)

$$
\begin{aligned}
A &\rightarrow B :< A, \{N_A\}_{K_B} > \\
B &\rightarrow A :< B, \{< N_A, N_B >\}_{K_A} > \\
A &\rightarrow B :< A, \{N_B\}_{K_B} >
\end{aligned}
$$

Deux rôles A(lice), et B(ob).

Protocole d'authentification (Needham-Schroeder)

$$
\begin{aligned}
A &\rightarrow B :< A, \{N_A\}_{K_B} > \\
B &\rightarrow A :< B, \{< N_A, N_B >\}_{K_A} > \\
A &\rightarrow B :< A, \{N_B\}_{K_B} >
\end{aligned}
$$

Keys : clefs de chiffrement K_A, K_B publiques (pour NS) ou privées.

Nonce : nombres aléatoires générés à chaque session N_A, N_B.

Messages : expressions construites avec constantes, paire $< \_, \_ >$, encryption $\{\_\}_\_$, hashing, etc.

Protocole d'authentification (Needham-Schroeder)

$$
\begin{aligned}
A &\rightarrow B :< A, \{N_A\}_{K_B} > \\
B &\rightarrow A :< B, \{< N_A, N_B >\}_{K_A} > \\
A &\rightarrow B :< A, \{N_B\}_{K_B} >
\end{aligned}
$$

Sessions : les agents (participants $a, b, \ldots$) jouent les rôles du protocole. Possibilité de sessions en parallèle et un agent peut jouer le rôle A dans une session, le rôle B dans une autre,...

Protocole d'authentification (Needham-Schroeder)

$$
\begin{aligned}
A &\rightarrow B :< A, \{N_A\}_{K_B} > \\
B &\rightarrow A :< B, \{< N_A, N_B >\}_{K_A} > \\
A &\rightarrow B :< A, \{N_B\}_{K_B} >
\end{aligned}
$$

Propriété : le protocole assure que A sait qu'il parle à B (et réciproquement).

Principe de l'analyse de sécurité :

- Le protocole est connu.
- La cryptographie est supposé parfaite (nécessité de connaitre les clés secrètes pour décrypter).
- Tout message émis peut être intercepté.
- Nombre de sessions en parallèle : nombre fini donné (usuellement 3-4) ou non borné.

L'attaquant (Intrus)

- Attaquant passif.
  Il ne peut interagir avec le protocole et peut juste intercepter et analyse les messages qui ont circulé.
- Attaquant actif.
  Il contrôle le réseau : peut intercepter, détruire, falsifier les messages, être un agent du protocole (participant malhonnête).

Les Propriétés

- Secrecy : l'intrus arrive à connaitre certaines données.

  Exemple : protocole d'échange d'une clé symmétrique entre deux participants : Le secret est la clé symmétrique.

Les Propriétés

- Secrecy : l'intrus arrive à connaitre certaines données.

  Exemple : protocole d'échange d'une clé symmétrique entre deux participants : Le secret est la clé symmétrique.

- Authentification : prouver son identité.
  nécéssaire pour paiement electronique !
  Réduit à du secret.

Les Propriétés

- Secrecy : l'intrus arrive à connaitre certaines données.

  Exemple : protocole d'échange d'une clé symmétrique entre deux participants : Le secret est la clé symmétrique.
- Authentification : prouver son identité.
  nécéssaire pour paiement electronique !
  Réduit à du secret.
- Anonymity :
  ne pas retrouver l'identité de l'envoyeur.

Les Propriétés

- Secrecy : l'intrus arrive à connaitre certaines données.

  Exemple : protocole d'échange d'une clé symmétrique entre deux participants : Le secret est la clé symmétrique.

- Authentification : prouver son identité.
  nécéssaire pour paiement electronique !
  Réduit à du secret.

- Anonymity :
  ne pas retrouver l'identité de l'envoyeur.

- Non-repudiation :
  impossibilité de contester son accord (signature de contrat).

Needham-Schroder n'est pas sûr

Principe de l'attaque : man in the middle

- 3 principaux a, b honnêtes, c malhonnête,
- 2 sessions en parallèle

L'attaquant utilise un participant honnête pour décrypter ce qu'il ne peut lire.

$a \rightarrow c :< a, \{N_a\}_{K_c} >$

1ere règle, session 1, a joue A, c joue B

$c(a) \rightarrow b :< a, \{N_a\}_{K_b} >$

1ere règle, session 2, c joue as a pour b

$b \rightarrow c :< b, \{< N_a, N_b >\}_{K_a} >$

2eme règle, session 2, b reponds c

$c \rightarrow a :< c, \{< N_a, N_b >\}_{K_a} >$

2eme règle, session 2, c joue B, a joue A

$a \rightarrow c :< a, \{N_b\}_{K_c} >$

3eme règle, session 1, a joue A, c joue B

$c(a) \rightarrow b :< a, \{N_b\}_{K_b} >$

3eme règle, session 2, c joue a pour b

b pense parler à a.

Une correction possible :

$$\begin{aligned} \mathrm{A} &\rightarrow \mathrm{B} : \{< \mathrm{A}, \mathrm{N_A} >\}_{\mathrm{K_B}} \\ \mathrm{B} &\rightarrow \mathrm{A} : \{< \mathrm{B}, < \mathrm{N_A}, \mathrm{N_B} >>\}_{\mathrm{K_A}} \\ \mathrm{A} &\rightarrow \mathrm{B} : \{< \mathrm{A}, \mathrm{N_B} >\}_{\mathrm{K_B}} \end{aligned}$$

Le cryptage de l'identité empêche c de se faire passer pour a.

Les règles de déduction de Dolev-Yao

Système (DY) :

Construction :

$$\frac{x \quad y}{< x, y >} \qquad \frac{x \quad y}{\{x\}_y}$$

Decomposition :

$$\frac{< x, y >}{y} \qquad \frac{< x, y >}{x} \qquad \frac{\{x\}_y \quad y}{x}$$

Modélisent le pouvoir de déduction de l'Intrus.

$T \vdash s$ avec $T = \{t_1, \ldots, t_n\}$ si s peut être déduit par utilisation répétée des règles de déduction.

Preuve de $T \vdash s$: suite d'application des règles teminant avec s.

Exemple

Hypothèses $\{N_A\}_{K_B}, K_A, K_B, N_B$

$$\dfrac{\dfrac{\dfrac{\{N_A\}_{K_B} \qquad K_B}{N_A} \qquad N_B}{< N_A, N_B >} \qquad K_A}{\{< N_A, N_B >\}_{K_A}}$$

Theorem

Les règles de déduction (DY)sont locales.

Local=si $T \vdash s$ alors il existe une preuve Pi n'utilisant que des sous-termes de $T \cup \{s\}$.

Conséquence : existence d'un algorithme polynomial pour savoir si $T \vdash s$.

Preuve de la localité

Récurrence sur la longueur de la preuve Π.

Cas 1 : la dernière règle est une construction.

$$\frac{\dfrac{\Pi_1}{u} \quad \dfrac{\Pi_2}{v}}{< u, v >}$$

Par hypothèse de récurrence la propriété est vraie pour $\dfrac{\Pi_1}{u}$ et $\dfrac{\Pi_1}{v}$, donc est vraie pour la preuve totale ($s =< u, v >$).

idem pour cryptage.

Preuve de la localité

Cas 2 : la dernière règle est une décomposition.

$$\frac{\dfrac{\Pi_1}{\{u\}_v} \quad \dfrac{\Pi_2}{v}}{u}$$

La dernière règle de Π_1 ne peut être une construction (sinon Π non minimale).

$\dfrac{\Pi_1}{\{u\}_v}$ de la forme $\dfrac{\dfrac{\Pi_1'}{< x, \{u\}_v >}}{\{u\}_v}$ Par hypothèse de récurrence elle ne contient que des sous termes de $T \cup \{\{u\}_v\}$, donc $< x, \{u\}_v >$ sous-terme de T.

Theorem

L'existence d'une attaque par un attaquant passif pour un nombre fini de sessions est décidable en temps polynomial.

Conséquence directe de la localité du système (DY).

La sécurité du protocole est décidable pour un nombre fini de sessions : un algorithme NP-complet pour trouver une attaque existe.

- Beaucoup plus difficile !
- Utilise la localité.
- Résolution de systèmes de contraintes sur des termes (expression avec les opérateurs, clés, variables).

Réécriture de Needham-Schroeder :

$$\begin{array}{llllllll} A & 0 & \rightarrow & \{N_A\}_{K_B} & B & \{y\}_{K_B} & \rightarrow & \{< y, N_B >\}_{K_A} \\ & \{< N_A, x >\}_{K_A} & \rightarrow & \{x\}_{K_B} & & \{N_B\}_{K_B} & \rightarrow & 0 \end{array}$$

- Choisir un nombre de sessions
- Choisir un entrelacement des actions de chaque sessions
- Exécution $u_0 \rightarrow v_0, u_1 \rightarrow v_1, \ldots, u_n \rightarrow v_n$
- Connaissance de l'intrus :
  - Initiale : $t_0, \ldots, t_p$
  - S'acroit de v_i à chaque exécution $u_i \rightarrow v_i$.

Système de contraintes associé à une exécution

$$
\begin{array}{lll}
(S) & t_0, \ldots, t_p & \models u_0 \\
 & v_0, t_0, \ldots, t_p & \models u_0 \\
 & \ldots & \\
 & v_{i-1} \ldots, v_0, t_0, \ldots, t_p & \models u_i \\
 & v_i, v_{i-1} \ldots, v_0, t_0, \ldots, t_p & \models u_i \\
 & \ldots & \\
 & v_n, \ldots, v_0, t_0, \ldots, t_p & \models s(\text{secret a trouver})
\end{array}
$$

σ solution de (S) ssi c'est une affectation des variables de (S) telle que $\sigma(v_i, \ldots, v_0, t_0, \ldots, t_p) \vdash \sigma(u_i),\ i = 0, \ldots, n.$

($\vdash$ déduction dans le système de DY)

Exemple : vérifier que le système de contraintes associé à une exécution normale de NS a une solution :

$$
\begin{array}{lcl}
0 & \models & 0 \\
\{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \models & \{\mathrm{y}\}_{\mathrm{K_B}} \\
\{<\mathrm{y}, \mathrm{N_B}>\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \models & \{<\mathrm{N_A}, \mathrm{x}>\}_{\mathrm{K_A}} \\
\{\mathrm{x}\}_{\mathrm{K_B}}, \{<\mathrm{y}, \mathrm{N_B}>\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \models & \{\mathrm{N_B}\}_{\mathrm{K_B}} \\
0, \{\mathrm{x}\}_{\mathrm{K_B}}, \{<\mathrm{y}, \mathrm{N_B}>\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \models & \text{pas de secret}
\end{array}
$$

Solution ?

Exemple : vérifier que le système de contraintes associé à une exécution normale de NS a une solution :

$$
\begin{array}{lcl}
0 & \models & 0 \\
\{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \models & \{\mathrm{y}\}_{\mathrm{K_B}} \\
\{<\mathrm{y}, \mathrm{N_B}>\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \models & \{<\mathrm{N_A}, \mathrm{x}>\}_{\mathrm{K_A}} \\
\{\mathrm{x}\}_{\mathrm{K_B}}, \{<\mathrm{y}, \mathrm{N_B}>\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \models & \{\mathrm{N_B}\}_{\mathrm{K_B}} \\
0, \{\mathrm{x}\}_{\mathrm{K_B}}, \{<\mathrm{y}, \mathrm{N_B}>\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \models & \text{pas de secret}
\end{array}
$$

Solution ? $\mathrm{y} = \mathrm{N_A}, \mathrm{x} = \mathrm{N_B}$

Exemple : vérifier que le système de contraintes associé à une exécution normale de NS a une solution :

$$
\begin{array}{lcl}
0 & \vdash & 0 \\
\{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \vdash & \{\mathrm{N_A}\}_{\mathrm{K_B}} \\
\{< \mathrm{N_A}, \mathrm{N_B} >\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \vdash & \{< \mathrm{N_A}, \mathrm{N_B} >\}_{\mathrm{K_A}} \\
\{\mathrm{N_B}\}_{\mathrm{K_B}}, \{< \mathrm{N_A}, \mathrm{N_B} >\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \vdash & \{\mathrm{N_B}\}_{\mathrm{K_B}} \\
0, \{\mathrm{N_B}\}_{\mathrm{K_B}}, \{< \mathrm{N_A}, \mathrm{N_B} >\}_{\mathrm{K_A}}, \{\mathrm{N_A}\}_{\mathrm{K_B}}, 0 & \vdash & \text{pas de secret}
\end{array}
$$

Forme réduite d'un système de contrainte : que des variables à droite de $\models$.

Theorem

Une forme réduite a une solution.

Theorem

Il existe un ensemble de règles de transformation qui transforme tout système en un système équivalent qui est soit $\bot$ (pas de solutions), soit un système en forme réduite.

(penser à la mise en forme triangulaire d'un système d'équations linéaires)

Et donc :

Theorem (Rusinowitch-Turuani)

Le problème de la sécurité des protocoles pour un nombre fini de sessions est Co-NP i.e. trouver une attaque est un problème NP (et même NP complet).

Nombre de sessions indéterminé : le problème est indécidable (il ne peut pas exister d'algorithme pour le résoudre).

Prise en compte de propriétés algébriques

- ou exclusif $\oplus$: associatif-commutatif, $x \oplus x = 0$,
- Homomorphisme $h(< x, y >) =< h(x), h(y) >$,
- ...

Approche similaire : montrer la localité, puis définir un algorithme de résolution de contraintes.

Un site de recensement de protocoles et d'attaques :

`www.lsv.ens-cachan.fr/spore`

Présentation inspirée de celle d'Hubert Comon, LSV, ENS-Cachan, (Cimpa school, Feb 2005)

Félicitations pour avoir tenu jusqu'ici ! ! !

Félicitations pour avoir tenu jusqu'ici ! ! !

Questions ?