Journées GDR-ALP

4 avril 2003, Marne-la-vallée

Cryptologie

F. Morain

Slide 1

http://www.lix.polytechnique.fr/Labo/Francois.Morain/

Plan

Slide 2

0. La problématique de la cryptologie moderne.

I. Pourquoi des protocoles ?

II. Protocoles sans authentification.

III. Identification.

IV. Échange de clefs avec authentification.

Slide 3

0. La problématique de la cryptologie moderne

Problème difficile → Cryptosystème avec sécurité élémentaire précisée → Preuve de sécurité (CCA, etc.) → Protocole → Preuve du protocole → Normalisation (IEEE, ANSI) → Brevets ; produits : SSL, etc.

But ultime : définir des modèles de sécurité et prouver des théorèmes.

Les besoins

Slide 4

– Intégrité : **empêcher la modification** d'un message.
– Confidentialité : **protéger** l'information.
– Signature : **c'est bien moi qui ait chiffré le message**.
– Authentification : **prouver qui on est**.

Plan

1. **Daniel Augot** : chiffrement, signature.
2. **FM** : quelques protocoles.
3. **Frédéric Blanqui** : introduction à la preuve de protocole.

I. Pourquoi des protocoles ?

Protocole : échange de communications et d'actions entre plusieurs acteurs.

Ex. sécurité à distance (login, commerce, etc.).

Slide 5 *Problématique :* comment suis-je sûr de l'identité de l'autre ? Que les calculs que je fais avec lui sont légitimes ?

L'authentification est le point dur de la crypto !

Deux problèmes : confiance initiale ; transfert de la confiance.

Comme toujours : définir les modèles de sécurité, le cahier des charges, et qui sont les méchants ?

Version idéale de l'univers : tout acteur a un couple $(K_{\mathsf{pub}}, K_{\mathsf{priv}})$; la clef publique est stockée dans un annuaire **accessible universellement**.

Mais :

Slide 6

- Qui maintient l'annuaire ? (ajouts, retraits, etc.)
- Comment accède-t-on à l'annuaire de manière sécurisée ?
- Comment diffuser cet annuaire ? Comment s'assurer que les copies sont valides ?
- Cela ne suffit pas : il faut un accord préalable off-line entre le détenteur de l'annuaire et l'acteur.

Cas où cela marche à peu près : réseaux locaux ou propriétaires (GSM), dans lesquels une autorité supérieure existe (Ca, TPC/TTP, etc.).

Méfions-nous des modèles simplistes : réseaux de confiance (web of trust), confiance globable, INTERNET, etc.

Slide 7

II. Échange de clef sans authentification

Réseaux captifs ; on craint juste l'espionnage passif.

A) Protocoles sans serveur sûr

Prérequis : Alice et Bob partagent une clef symétrique commune K.

Avec une seule passe :

Protocole : $A \to B \ : E_K(r_A, t_A, B)$

Actions : Alice envoie à Bob $E_K(r_A)$, la clef de session est $W = r_A$.

Propriétés : authentification de W, fraicheur ; mais Alice "force" la clef de Bob.

Avec deux passes :

Protocole :

$$\begin{array}{ccccll} A & \to & B & : & E_K(r_A, t_A, B) & (1) \\ A & \leftarrow & B & : & E_K(r_B, \mathcal{H}(r_A, r_B)) & (2) \end{array}$$

Actions : la clef de session est $\mathcal{H}(r_A, r_B)$.

Propriétés : Alice et Bob ne peuvent prévoir la clef.

Le protocole de Shamir

Slide 8

Présentation imagée :

- Alice met la clef dans un coffre fermé avec son cadenas et envoie le coffre à Bob ;
- Bob rajoute son cadenas et renvoie le coffre à Alice ;
- Alice enlève son cadenas et renvoie le coffre à Bob ;
- Bob ouvre le coffre.

Rem. on peut généraliser à la distribution de cartes par la poste.

Réalisation pratique : p tq LD soit impossible à résoudre dans $(\mathbb{Z}/p\mathbb{Z})^*$.
- Alice choisit $a \in_R (\mathbb{Z}/(p-1)\mathbb{Z})^*$ et envoie $x = M^a \bmod p$;
- Bob choisit $b \in_R (\mathbb{Z}/(p-1)\mathbb{Z})^*$ et calcule $y = x^b \bmod p$ et l'envoie à Alice ;
- Alice calcule $z = y^{a^{-1}} = M^b$ et l'envoie à Bob ;
- Bob calcule $z^{b^{-1}} = M$.

B) Protocoles avec un serveur sûr

Le protocole de Needham et Schroeder

Prérequis : A et T ont une clef commune K_{AT} ; B et T ont K_{BT} ;

Protocole :

Slide 9

$$\begin{array}{llcll} A & \rightarrow & T & : & A, B, n_A & (1) \\ A & \leftarrow & T & : & E_{K_{AT}}(n_A, B, k, E_{K_{BT}}(k, A)) & (2) \\ A & \rightarrow & B & : & E_{K_{BT}}(k, A) & (3) \\ A & \leftarrow & B & : & E_k(n_B) & (4) \\ A & \rightarrow & B & : & E_k(n_B - 1) & (5) \end{array}$$

Propriétés : authentification de A pour B (messages (4) et (5)), établissement de la clef commune k avec confirmation.

Pb. Bob ne peut être sûr de la fraicheur de la clef k. Protocole amélioré utilisé dans Kerberos (avec timbre-poste).

III. Identification

A) à distance

Slide 10

Idée (vraiment très) naïve : stocker les mots de passe en clair sur Oscar.

Mieux : on ne stocke pas le mot de passe, mais $F(m)$ où F est difficile à inverser ; on recalcule à chaque fois $F(m)$. En `UNIX`, DES modifié. **Mais :** les mots de passe transitent en clair... !

Idée : chercher à s'authentifier via la possession d'un secret, qui n'est jamais divulgué sur les ondes.

Prérequis : au moment de la création du compte, Alice obtient une clef de chiffrement commune avec Oscar.

Slide 11

Challenge/réponse

Protocole intuitif :

$$\begin{array}{llll} O & \leftarrow & A : & \text{je veux me connecter} \\ O & \rightarrow & A : & r_O \\ O & \leftarrow & A : & E_K(r_O, r_A) \\ O & \rightarrow & A : & r_A \end{array}$$

Mais **attaque par réflexion :**

$$\begin{array}{rc} O \rightarrow E : & r_O \\ \mathbf{O \leftarrow E} : & \mathbf{r_O} \\ \mathbf{O \rightarrow E} : & \mathbf{E_K(r_O, r'_O)} \\ O \leftarrow E : & E_K(r_O, r'_O) \\ O \rightarrow E : & r'_O \end{array}$$

Oscar et Alice croient s'être authentifiés mutuellement. . .

Prouver qu'on détient une clef commune ne suffit donc pas.

Le protocole est de plus trop symétrique.

Slide 12

Une meilleure solution

Prérequis : Alice possède un algorithme de signature S_A et Oscar détient l'algorithme de vérification V_A.

1. Oscar envoie un nombre aléatoire r à Alice ;
2. Alice, **en local**, signe r et retourne $S_A(r)$;
3. Oscar vérifie que $V_A(S_A(r)) = r$.

On évite l'attaque par réflexion grâce à la dissymétrie.

Pb : que veut dire local en Unix ? (environnement multi-utilisateur, stockage des clefs privées, etc.).

Plus évolué : SSH

But : remplacer `rlogin` par un tunnel sécurisé (`slogin`).

Prérequis : Alice possède une copie de la clé publique Pub_M d'Oscar (pour éviter les *man-in-the-middle*).

Protocole :

Slide 13

- Alice envoie une requête d'authentification à Oscar ;
- Oscar renvoie Pub_M, ainsi qu'une clef publique de session Pub_s ;
- Alice vérifie que la clef publique est la bonne ;
- Alice génère une sous-clef de session k_s pour utilisation dans un algorithme symétrique et renvoie $Pub_M(Pub_s(k_s))$ à Oscar ;
- Alice et Oscar correspondent avec k_s : ils commencent par une authentification par mot de passe classique – chiffré – (comme par rlogin), ou de type RSA : le client déchiffre un challenge chiffré par le serveur et renvoie le haché.

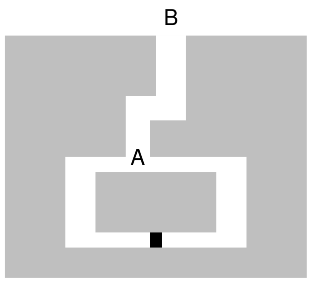
B) En environnement contraint

Typiquement : carte à puce vis-à-vis d'un lecteur.

Preuves à divulgation nulle (zero-knowledge) : prouver son identité sans révéler de secret.

Réponse : le paradigme de la caverne (Guillou–Quisquater).

Slide 14



Exemple : Fiat-Shamir

Prérequis : Alice choisit deux nombres premiers secrets p et q, forme $N = pq$, $s \in_R [1, N[$ et calcule $v = s^2 \bmod N$.

Clef publique : (N, v).

Clef privée : s.

Slide 15

Protocole : Bob veut vérifier l'identité d'Alice, par exemple en étant convaincu que Alice possède p et q.

1. Alice choisit $r \in_R [1, N[$ et envoie $x = r^2 \bmod N$ à Bob.
2. Bob envoie un bit aléatoire b ($b = 0$ ou 1) à Alice.
3. Si $b = 0$, Alice renvoie $y = r$; si $b = 1$, elle renvoie $y = rs \bmod N$.
4. Si $b = 0$, Bob vérifie que $y^2 \equiv x \bmod N$ (Alice connaît $\sqrt{x}$) ; si $b = 1$, Bob teste si $y^2 = xv \bmod N$ (Alice connaît $\sqrt{v}$).

IV. Échange de clefs avec authentification

A) Needham-Schroeder

Prérequis : A et B disposent des clefs publiques authentiques de l'autre ;

Protocole :

Slide 16

$$\begin{array}{ccccll} A & \rightarrow & B & : & E_B(A, N_A) & (1) \\ A & \leftarrow & B & : & E_A(N_A, N_B) & (2) \\ A & \rightarrow & B & : & E_B(N_B) & (3) \end{array}$$

Propriétés : quand A déchiffre (2), elle vérifie que N_A est la bonne valeur et en déduit que B connait N_A et est bien authentifié. B fait de même avec (3). La clef de session est $f(N_A, N_B)$ avec f non inversible.

mais... cf. exposé de FB.

Slide 17

B) L'algorithme de Diffie-Hellman

Prérequis : p nombre premier, g donnés.

Protocole :

$$\begin{array}{ccccc} A & \rightarrow & B & : & g^a \\ A & \leftarrow & B & : & g^b \end{array}$$

Actions : à la fin de l'échange, A et B possède g^{ab}.

Propriétés : la sécurité est basée sur le problème DH.

Attaque au milieu : (*Mafia attack* ou *man-in-the-middle*).

Alice		Charlie		Bob
g^a	$\rightarrow$	g^c	$\rightarrow$	
	$\leftarrow$	g^c	$\leftarrow$	g^b
g^{ac}		g^{ac}, g^{bc}		g^{bc}

Station-to-station protocol

Prérequis : chaque utilisateur a une clef publique (p, g) un générateur de $(\mathbb{Z}/p\mathbb{Z})^*$, $2 \leqslant g \leqslant p-2$; un couple (S, V).

Protocole :

Slide 18

$$\begin{array}{ccccll} A & \rightarrow & B & : & g^x \bmod p & (1) \\ A & \leftarrow & B & : & g^y \bmod p, E_k(S_B(g^y, g^x)) & (2) \\ A & \rightarrow & B & : & E_k(S_A(g^x, g^y)) & (3) \end{array}$$

Rem : $k = g^{xy}$;

Propriétés : échange de clefs et authentification mutuelle des acteurs et de la clef.

Un protocole réel : RNIS

RNIS : Réseau Numérique à Intégration de Service ; utilisé dans les GSM.

Prérequis : chaque téléphone T a un couple $(Pub_T, Priv_T)$ ainsi que $\mathcal{C}(T) = S_G(Pub_T, \#_T, \text{position}_T)$.

Slide 19

Quand T_A veut communiquer avec T_B :

- T_A et T_B échangent une **clef de session** aléatoire k_s à la Diffie-Hellman ; k_s est une clef DES qui sert à tous les échanges subséquents ;
- T_A et T_B s'échangent $\mathcal{C}(T_A)$ et $\mathcal{C}(T_B)$;
- T_A (resp. T_B) vérifie la signature et en déduit Pub_{T_B} (resp. Pub_{T_A}) ;
- chaque téléphone envoie des **messages tests** (dépendant du temps) que l'autre doit signer en temps réel ; si les tests sont corrects, la communication est établie ;
- à la fin de la communication, k_s **est détruite**.

Conclusions

Les protocoles d'échange de clefs, ce n'est pas simple, toute **aide** est la bienvenue.

Autres protocoles : cryptographie distribuée ; partage de secrets ; cryptographie à seuil, etc.

Slide 20