COURS ID12 « INTRODUCTION À LA CRYPTOLOGIE »
COURS 7
CRYPTOGRAPHIE ASYMÉTRIQUE : ATTAQUES SUR RSA ET PROTOCOLES

PAUL ZIMMERMANN

2. RSA en détail (suite et fin)

2.1. **Attaque cyclique.** Soit $c = m^e \bmod n$ le message chiffré correspondant au message clair m. Supposons que l'on trouve un entier k tel que $c^{e^k} \equiv c \bmod n$. Soit alors $m' = c^{e^{k-1}} \bmod n$. On a $m'^e \equiv c^{e^k} \equiv c \bmod n$, donc $m'^e - m^e \equiv 0 \bmod n$. Cela implique soit $m' = m$, sinon $m' - m$ est un diviseur non trivial de n. Bref, dans tous les cas on peut retrouver m à partir de m'. On parle alors d'attaque cyclique (*cycling attack* en anglais).

2.2. **Attaque par $P - 1$.** Supposons que $q - 1$ soit B-friable, c'est-à-dire $q - 1 = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ avec $p_1 < p_2 < \cdots < p_k$, où $p_j^{e_j} < B$, pour B « petit ». Soit K le produit de tous les p^e pour p premier inférieur à B, et e l'exposant maximal tel que $p^e < B$. Alors K est un multiple de $q - 1$, soit $K = \lambda(q-1)$, donc $c^K \equiv c^{\lambda(q-1)} \bmod q$. Par le petit théorème de Fermat, on a $x^{q-1} \equiv 1 \bmod q$ pour tout x, donc $c^K \equiv 1 \bmod q$. On en déduit que $\gcd(c^K - 1, n) = q$, sauf si $p - 1$ est aussi B-friable, auquel cas $\gcd(c^K - 1, n) = n$, et il suffit (par exemple par dichotomie) de diminuer la borne B pour séparer p et q.

Le produit K est énorme, mais en fait il n'est pas nécessaire de le calculer. L'algorithme correspondant est le suivant :

Algorithme $P - 1$.
Choisir c aléatoire, $0 < c < n$
$p \leftarrow 2$
while $p < B$ do
    $q \leftarrow p$
    while $q < B$ do
        $c \leftarrow c^p \bmod n$
        $q \leftarrow qp$
    end
end
Return $\gcd(c - 1, n)$.

Cet algorithme permet en fait de trouver les facteurs p de n tels que $p - 1$ soit B-friable. Un algorithme similaire existe pour les p tels que $p + 1$ est B-friable.

2.3. **Exemples historiques.**

Date: `zimmerma@loria.fr`.

2.3.1. *RSA-100.* $p = 40094690950920881030683735292761468389214899724061$

$p - 1 = 2^2 \cdot 5 \cdot 41 \cdot 2119363 \cdot 602799725049211 \cdot 38273186726790856290328531$

$38273186726790856290328531 - 1 = 2 \cdot 3 \cdot 5 \cdot 61 \cdot 113 \cdot 93557 \cdot 1978284752702551$

$p + 1 = 2 \cdot 3 \cdot 11 \cdot 59 \cdot 1029653080403720622256901265864444886804031773$

$q = 37975227936943673922808872755445627854565536638199$

$q - 1 = 2 \cdot 3167 \cdot 3613 \cdot 587546788471 \cdot 3263521422991 \cdot 865417043661324529$

$865417043661324529 - 1 = 2^4 \cdot 3 \cdot 11 \cdot 17 \cdot 61 \cdot 1580566471723$

$q + 1 = 2^3 \cdot 3 \cdot 5^2 \cdot 109 \cdot 409 \cdot 20839813 \cdot 60236089 \cdot 49147216823 \cdot 23011759155976667$

2.3.2. *Autres "challenges" RSA.* RSA-140 a été factorisé le 2 février 1999 (5.4% de l'étape de crible fut réalisée au LORIA/INRIA Lorraine). Le crible nécessita de l'ordre de 60 PCs à 300 Mhz.

RSA-155 a été factorisé le 22 août 1999, 4.5% du crible fut réalisé avec les machines du Centre Charles Hermite.

RSA-576 (174 chiffres) a été factorisé le 3 décembre 2003 par Franke et Kleinjung.

RSA-200 a été factorisé le 9 mai 2005 par Bahr, Boehm, Franke, Kleinjung. C'est le record actuel. Le crible a pris de l'ordre de 55 années d'Opteron 2.2 Ghz.

Voir `http://www.rsasecurity.com/rsalabs/node.asp?id=2093` pour en savoir plus. Le prochain nombre à factoriser est RSA-704 (212 chiffres), pour lequel un prix de 30.000 dollars est offert. Pour RSA-2048 (617 chiffres), un prix de 200.000 dollars est offert.

3. Protocoles d'authentification

Référence : chapitre 10 de [1].

3.1. **Mots de passe.** Mots de passe stockés en clair ou hachés (par fonction à sens unique, en pratique pour des raisons historiques on utilise un procédé de chiffrement). Empêche le piratage du mot de passe, mais n'évite pas le rejeu.

Utilisation d'une fonction à sens unique pour vérifier un mot de passe :

Ici vient la figure 10.1 page 390 de [1].

Remarque 1 : pour ralentir les attaques essayant un grand nombre de mots de passe, on peut rendre le calcul de la fonction à sens unique coûteux, par exemple en multipliant le nombre d'étapes (RSA ou DES).

Remarque 2 : pour éviter les attaques à base de dictionnaire, on ajoute une composante aléatoire de t bits (*salt* en anglais), qui augmente de 2^t la difficulté d'une attaque par dictionnaire.

Attaques : (1) rejeu, (2) recherche exhaustive, (3) attaque par dictionnaire.

3.1.1. *Mots de passe Unix.*

Ici vient la figure 10.2 page 394 de [1].

3.1.2. *One-time password.*

Algorithm OneTimePassword (Lamport)
I_1. User A chooses a secret w, a one-way function H, and a constant t
I_2. A transfers $w_0 = H^t(w)$ to B, and B sets $i_A = 1$
A_i. A sends to B the message $A, i, w_i = H^{t-i}(w)$
V_i. B checks that $i = i_A$, $H(w_i) = w_{i-1}$, and $i_A \leftarrow i_A + 1$.

3.2. **Protocoles à défi.**

3.2.1. *Avec chiffrement à clé secrète.* Authentification unilatérale avec estampille (l'astérisque dénote les champs optionnels) : B vérifie que l'estampille t_A est correcte (évite le rejeu).

$$A \to B : E_K(t_A, B^*).$$

Authentification unilatérale avec nombre aléatoire (évite une estampille, avec un message de plus) :

$$\begin{array}{rcl} A & \leftarrow & B : r_B \\ A & \rightarrow & B : E_K(r_B, B^*). \end{array}$$

Authentification bilatérale avec nombre aléatoire :

$$\begin{array}{rcl} A & \leftarrow & B : r_B \\ A & \rightarrow & B : E_K(r_A, r_B, B^*) \\ A & \leftarrow & B : E_K(r_B, r_A). \end{array}$$

Remarque : on peut remplacer E_k par une fonction de hachage avec clé h_k (*message authentification code* ou MAC en anglais) :

$$\begin{array}{rcl} A & \leftarrow & B : r_B \\ A & \rightarrow & B : r_A, h_K(r_A, r_B, B) \\ A & \leftarrow & B : h_K(r_B, r_A, A). \end{array}$$

3.3. **Protocoles sans divulgation d'information.** Plus connus sous le nom de *zero knowledge* en anglais.

Problème avec les mots de passe : quand A (*prover*) donne son mot de passe à B (*verifier*), B peut ensuite se faire passer pour A. Les protocoles à défi résolvent en partie ce problème : l'information donnée à B n'est pas directement réutilisable, mais donne des informations partielles sur le secret de A.

Structure générale. Trois passes : gage (*witness*), défi (*challenge*), réponse.

Protocole de Fiat-Shamir
I_1. T choisit $n = pq$
I_2. A choisit $1 \leq s \leq n-1$ premier avec n, calcule $v = s^2 \bmod n$, enregistre v auprès de T
A_1. $A \to B : x = r^2 \bmod n, \quad 1 \leq r < n$
A_2. $A \leftarrow B : e \in \{0, 1\}$
A_3. $A \to B : y = r \cdot s^e \bmod n$
V_1. B vérifie $y^2 = x \cdot v^e \bmod n$

Si Charlie choisit y au hasard, et prend $x = y^2/v$, cela répond correctement si $e = 1$; mais si $e = 0$, il faut trouver $x^{1/2} \bmod n$: sécurité basée sur la difficulté de l'extraction de racine carrée modulo n.

Autres protocoles : Feige-Fiat-Shamir (basé aussi sur racine carrée), Guillou-Quisquater (GQ, basé sur problème RSA, i.e. extraction de racine v-ème modulo n), Schnorr (basé sur logarithme discret).

3.4. **Attaques.** Rejeu. Solutions : protocoles à défi, composantes aléatoires (*nonces*), ...

Entrelacement : C se place entre A et B, et pose à A les questions de B. Solution : chaîner les messages d'un même protocole (*chained nonces*).

Texte choisi : dans un protocole à défi, Charlie choisit les défis pour extraire de l'information du secret de A. Solutions : protocoles sans divulgation d'information.

Références

1. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997, freely available at `http ://www.cacr.math.uwaterloo.ca/hac/`.