Preuves de sécurité dans le modèle de l'oracle aléatoire

1. Oracle aléatoire

De nombreux schémas cryptographiques utilisent des fonctions de hachage et/ou des générateurs pseudo-aléatoires. Typiquement, ces objets sont utilisés en pratique en chiffrement à clé publique et en signature pour introduire de la redondance et contrer des attaques à messages choisis. Leur utilisation, un peu empirique autrefois, a attiré l'attention des chercheurs désireux de préciser leur rôle dans le renforcement de la sécurité.

Concrètement, ces objets sont des applications de $\{0,1\}^k$ dans $\{0,1\}^l$ et ce que l'on leur demande, plus généralement que les propriétés usuelles des fonctions de hachage (sens-unique, résistance aux collisions), c'est que la connaissance des images de certains éléments de $\{0,1\}^k$ ne soit en pratique pas exploitable pour prévoir les images d'autres éléments.

Le *modèle des oracles aléatoires* consiste à prendre cela comme hypothèse de travail, de façon un peu radicale, c'est-à-dire du point de vue de la théorie de l'information : la connaissance des images de certains points de $\{0,1\}^k$ n'apporte **aucune** information sur les images des autres points. On suppose aussi, qu'avant son utilisation, l'oracle est totalement inconnu. Autrement dit, son entropie est maximale ($l2^k$) dans l'ensemble des applications de $\{0,1\}^k$ dans $\{0,1\}^l$. Puisque les images de chaque point sont indépendantes, chaque réponse de l'oracle (appelé avec un nouvel argument) fait baisser cette entropie d'exactement l bits.

Les preuves dans ce modèle consistent à utiliser un éventuel attaquant sur le schéma que l'on veut prouver sûr et à en déduire un algorithme supposé ne pas exister (typiquement, un inverseur pour une fonction à sens unique). L'attaquant a la possibilité d'interroger les oracles à volonté mais on suppose (l'ai-je assez répété ?) que l'information qu'il possède sur ces oracles est strictement limité aux réponses qu'il obtient à ces occasions.

2. Sécurité en chiffrement

Sécurité sémantique

On considère une famille de cryptosystèmes indexée par un paramètre de sécurité (ici, la taille n des clés). Un attaquant (à clairs choisis) de la sécurité sémantique est un paire $A = (A_1, A_2)$ d'algorithmes destinés à être utilisés selon le schéma suivant :

On choisit une paire de clés de taille n au hasard.
A_1 génére deux clairs m_0 et m_1, et s une liste d'informations qu'il souhaite transmettre à A_2.
On choisit $b \in \{0,1\}$ au hasard (non communiqué à A).
On calcule un chiffré c de m_b.
A_2 calcule b' à partir de m_0 et m_1, de c et de s (et des clés).

On dit que cet attaquant est *d'avantage* ε (une fonction de n) si la probabilité que $b' = b$ dans le schéma ci-dessus est $\geqslant 1/2 + \varepsilon$. On dit que la famille de cryptosystèmes est *sémantiquement sure* s'il n'existe pas attaquant polynomial d'avantage ε sauf pour des fonctions ε négligeables.

Schéma de Bellare & Rogaway

Une fonction f à sens unique avec trappe (RSA ou Rabin, par exemple) nous permet de construire un cryptosystème selon le schéma

$$c = f(m) \qquad m = f^{-1}(c).$$

Mais ce cryptosystème souffre de nombreuses faiblesses. En particulier, il n'est pas sémantiquement sûr, même sous une attaque à messages clairs.

Bellare et Rogaway ont introduit le schéma suivant et ont montré qu'il est sémantiquement sûr, relativement à une attaque à clairs choisis, dans le modèle de l'oracle aléatoire. Par f, on désigne une famille de fonctions à sens unique avec trappe (correspondant chacune à une paire de clés). L'expéditeur calcule un chiffré c d'un message m par :

$$c := F(m,r) = \big(f(r), G(r) \oplus m\big) \qquad \text{où } r \in \{0,1\}^k \text{ est choisi au hasard.}$$

Le destinataire déchiffre $c = (c_1, c_2)$ par

$$m := G\big(f^{-1}(c_1)\big) \oplus c_2.$$

Preuve de sécurité

Pour montrer la sécurité annoncée, on suppose l'existence d'un attaquant A de la sécurité sémantique, d'avantage ε non-négligeable et utilisant la fonction G comme un oracle aléatoire. On construit alors un inverseur B de la fonction f ayant une probabilité de succès non-négligeable. Le code de B est décrit ci-dessous. On l'applique après avoir choisi une paire de clés de taille n, un $x_0 \in \{0,1\}^k$ au hasard (gardé secret), et avoir posé $y = f(x_0)$.

B

Entrée : un élément y de $\{0,1\}^k$.
Sortie : un élément x de $\{0,1\}^k$ (tel que $y = f(x)$ avec probabilité non-négligeable).

Exécuter A_1 pour obtenir m_0, m_1 et s.
$c_1 := y$
Choisir c_2 au hasard.
Exécuter A_2 avec les données m_0, m_1, $c = (c_1, c_2)$ et s.
Rechercher si, dans la liste des appels à G par A, un antécédent de y par f était l'argument.
Si oui, retourner cet argument. Si non, retourner une valeur aléatoire.

On ne fait pas fonctionner B (donc A) avec un oracle fixé à l'avance comme dans le cas réel, mais avec un oracle G', au départ non spécifié, et qui va être précisé au fur et à mesure des besoins. Le fait que cet oracle ne soit pas défini au départ est rendu possible par le modèle de l'oracle aléatoire. En effet, puisque l'information détenue au sujet de cet oracle dans ce modèle, se limite strictement aux réponses aux requêtes effectuées, le fait qu'il soit défini à l'avance ou seulement au cours de son utilisation est totalement transparent du point de vue de l'attaquant, pourvu que l'oracle reste uniformément aléatoire.

Le comportement de l'oracle G' au cours des différentes requêtes est le suivant. Pour chaque appel de G' avec un argument pour lequel il a déjà été appelé, il retourne le même argument que les fois précédentes (G' est bien une fonction). Lorsque G' est appelé avec un argument r présenté pour la première fois, sa valeur $G'(r)$ est calculée comme suit. Si $f(r) \neq y$ alors $G'(r)$ est pris au hasard. Si $f(r) = y$ alors $b \in \{0,1\}$ est choisi au hasard et on pose $G'(r) = c_2 \oplus m_b$. On notera que la fonction G' est uniformément aléatoire (avant que c_2 ne soit encore fixé), ce qui est nécessaire pour rester en conformité avec le modèle.

On remarque alors que, lorsque la réponse est "non" dans la dernière ligne de B, le chiffré c n'est lié en aucune façon ni à m_0 ni à m_1 et donc la réponse de l'attaquant dans ce cas est exacte avec probabilité $1/2$. L'hypothèse que l'attaquant A ait globalement un avantage ε implique que la réponse dans la dernière ligne de B soit "oui" avec probabilité $\geqslant \varepsilon$. Cela signifie que l'inverseur a une probabilité de succès non-négligeable, condredisant le fait que f soit à sens unique.

3. Sécurité en signature

Forge existentielle

On considère la sécurité vis-à-vis d'une forge existentielle adaptative à messages choisis. Dans ce cadre, on donne à l'attaquant la possibilité de chiffrer les messages de son choix à chaque moment (par l'intermédiaire d'un oracle de signature) et on s'intéresse à sa capacité à générer une signature valide d'un autre message de son choix (sans recours cette fois-ci à l'oracle). L'attaquant est donc représenté par un algorithme ayant accès à un oracle de signature (plus les oracles d'éventuelles fonctions de hachages utilisées) produisant la signature d'un message m avec la seule restriction que m ne soit pas soumis à l'oracle de signature. Le schéma de signature est sûr s'il n'existe pas de tel attaquant de coût polynomial et d'avantage non négligeable.

Signature *Full Domain Hash*

C'est le schéma classique de signature utilisant une permutation à sens unique $f : E \to E$ basée sur RSA ou Rabin (donc multiplicative) et une fonction de hachage H à valeurs dans E, mais avec l'hypothèse que le domaine des valeurs de H est E tout entier (potentiellement, en tant que fonction aléatoire).

On calcule donc la signature s d'un message m par :

$$s := f^{-1}\big(H(m)\big),$$

et on vérifie la signature par

$$f(s) \stackrel{?}{=} H(m).$$

Preuve de sécurité

On va montrer que ce schéma résiste à une forge existentielle à message choisis, dans le modèle de l'oracle aléatoire. (Cette preuve utilise la multiplicativité de f. Il existe une autre preuve, avec une réduction plus mauvaise, mais n'utilisant pas la multiplicativité). Pour cela, on suppose l'existence d'un attaquant A polynomial et ayant un taux de réussite ε non négligeable et on construit un inverseur B de f aussi polynomial et ayant un taux de réussite non négligeable. L'attaquant dispose d'un oracle S de signature et d'un oracle pour la fonction H. On peut supposer que l'attaquant n'appelle jamais l'oracle de signature sans avoir auparavant explicitement appelé l'oracle H avec le même argument.

Voici la description de l'inverseur :

B
- Entrée : un élément y de E.
- Sortie : un élément x de E (tel que $y = f(x)$ avec probabilité non-négligeable).

- Exécuter A, qui va retourner un couple (m, s).
- Appeler H' avec l'argument m (si cela n'a pas été fait auparavant).
- Retourner s/r, où r est la valeur utilisée lors du calcul de $H'(m)$.

Cet inverseur utilise l'oracle H' défini de la façon suivante (où p est une probabilité à fixer) :

H'

Entrée : Un élément m de E.

Si H' à déjà été appelé avec la valeur m, retourner la même valeur que précédemment.
Sinon
- Choisir r au hasard dans E.
- Retourner $f(r)$ (avec probabilité p)
- ou
- Retourner $yf(r)$ (avec probabilité $1 - p$).

Puisque f est bijective, cet oracle ainsi défini est bien une fonction uniformément aléatoire parmi les fonctions à valeurs dans E.

L'oracle de signature, quant à lui, retourne r lorsque l'oracle H' a auparavant retourné $f(r)$ pour le même argument, et provoque un échec de B lorsque H' a auparavant retourné $yf(r)$.

L'inverseur fournit un antécédent de y par f dès que les trois conditions suivantes sont réunies. (1) L'attaquant A retourne une signature valide. (2) L'oracle de signature ne provoque pas d'échec (il n'est jamais appelé avec des valeurs pour lesquelles H' a retourné $yf(r)$). (3) L'oracle H' a retourné une valeur de la forme $yf(r)$ pour l'argument m.

Si q est le nombre d'appels de l'attaquant à l'oracle de signature, le succès de l'inverseur est donc

$$\varepsilon' \geqslant \varepsilon p^q (1 - p).$$

On maximise le membre de droite avec la valeur $p = q/(q + 1)$. On obtient alors

$$\varepsilon' \geqslant \varepsilon \frac{q^q}{(q + 1)^{q+1}}.$$

Pour q grand, la minoration obtenue est environ ε/qe. Quantitativement, supposer que $\varepsilon' qe$ est négligeable (où ε' majore l'avantage des inverseurs de f) permet d'assurer la sécurité du schéma de signature *Full Domain Hash*.