Cryptographie – Feuille d'exercices 7

Cryptanalyse théorique du cryptosystème RSA

M1 Informatique – 2014-2015

1 Exercice : Malléabilité et Indistinguabilité

De manière générale, on souhaite que les messages chiffrés d'un cryptosystème aient une répartition aléatoire afin d'empêcher tout attaquant qui se contenterait d'observer les messages chiffrés de récupérer des informations. Cette propriété est capturée par la notion d'intistinguabilité de deux messages chiffrés : si étant donnés deux messages m_1 et m_2 choisis par un adversaire, celui-ci ne peut décider de quel message provient un chiffré c, on dit que le cryptosystème est *indistinguable*. De plus, il est souhaitable qu'un attaquant ne puisse pas obtenir un message chiffré valide en modifiant un message chiffré qu'il aurait intercepté. On dit alors que le cryptosystème n'est pas *malléable*.

1. Rappelez le fonctionnement du cryptosystème RSA ;
2. Montrer que ce cryptosystème est à la fois malléable et distinguable.

On propose alors une variante dite RSA - OAEP (Optimal Asymetric Encryption Padding) qui fonctionne de la manière suivante :
- On se donne deux tailles : n et k_r ;
- G est une fonction de hachage cryptographique dont la sortie est sur $n - k_r$ bits ;
- H est une fonction de hachage cryptographique dont la sortie est sur k_r bits ;
- On suppose que l'on souhaite chiffrer des messages de taille comprise entre 1 et $n - k_r$ bits (lorsque le message a une taille strictement inférieure à $n - k_r$, on le complète avec des 0 afin d'atteindre un bloc de taille $n - k_r$) ;
- On génère une valeur aléatoire r sur k_r bits ;
- Enfin, on chiffre enfin le message suivant :

$$[(m||000) \oplus G(r)] \,||[H\,((m||000) \oplus G(r)) \oplus r].$$

3. Montrer comment le destinataire peut déchiffrer un message ;
4. Quel intérêt y a-t-il selon vous à utiliser RSA - OAEP ?

2 Exercice : Factorisation

On suppose $n = p \times q$, où p et q sont deux nombres premiers distincts. On rappelle que $\varphi(n) = (p-1)(q-1)$.

1. On suppose que n et $\varphi(n)$ sont connus. Montrer que l'on peut alors retrouver p et q.
2. Trouver la factorisation de n dans les deux cas suivants :
   - $n = 667$, $\varphi(n) = 616$;
   - $n = 15049$, $\varphi(n) = 14800$.

3 Exercice : RSA avec un modulo commun

On suppose que deux entités Alice et Bob utilisent un schéma de chiffrement RSA avec le même modulo n et des exposants publics différents e_1 et e_2.

1. Montrer qu'Alice peut déchiffrer les messages adressés à Bob.
2. Montrer qu'un attaquant Charlie peut déchiffrer un message envoyé à la fois à Alice et Bob lorsque $\text{pgcd}(e_1, e_2) = 1$.

4 Exercice : Génération de clés RSA

Dans cet exercice, on montre que lors de la génération des clés RSA, on doit prendre garde au fait que $q - p$ ne soit pas trop petit, où $n = p \times q$ et $q > p$.

1. Supposons que $q - p = 2d > 0$ et $n = p \times q$. Montrer que $n + d^2$ est un carré parfait.
2. Étant donné un entier n, qui est le produit de deux nombres premiers impairs et, étant donné un petit entier d tel que $n + d^2$ soit un carré parfait ; montrer comment cette information peut être utilisée pour factoriser n.
3. Utiliser cette technique pour factoriser $n = 2189284635403183$.

5 Exercice : Chiffrement RSA itéré

Un des outils du cryptanalyste consiste à rechiffrer plusieurs fois le message chiffré. Il arrive que, pour un cryptosystème apparemment solide, on retrouve le message clair après un petit nombre d'applications de la fonction de chiffrement. Ceci constitue évidemment une grave faiblesse du schéma.

1. Soit $n = 35$ un modulo RSA, m le message clair et c le message chiffré. Vérifier que $E(c) = m^{e^2} = m$ pour tout exposant e légitime (c'est à dire tout entier e tel que $0 < e < \varphi(35)$ et $\text{pgcd}(e, \varphi(35)) = 1$), de sorte que le système n'est pas sûr.
2. Expliquer comment mettre en place une "attaque par cycle" de façon à déchiffrer un message chiffré c, lorsque l'on connaît la clé publique correspondante (n, e).
3. Essayer de trouver les conditions qui rendent l'attaque possible. Proposer une méthode de génération des paramètres RSA, qui permette d'empêcher cette attaque.
   Remarque : on peut montrer que la probabilité de succès d'une telle "attaque par cycle" est négligeable si p et q sont choisis aléatoirement et suffisamment grands.