# Corrigé du TD : Cryptographie

Nouha Oualha et Artur Hecker

*Oct. 2009*

## *Exercice 1 :*

Pour protéger les fichiers $\{F_i\}$ lors de leur transmission, Astrid et Béatrice doivent suivre plusieurs étapes :

1) Génération des paires de clés publique/privée (RSA) : Astrid et Béatrice doivent posséder des clés RSA qui vont les permettre de s'authentifier mutuellement. Elles peuvent générer ces clés individuellement (alternativement, elles peuvent avoir recours à une tierce de confiance dite générateur de clés privées).
2) Création des certificats : Astrid et Béatrice peuvent publier leurs clés publiques dans un annuaire public sécurisé et ainsi chacune peut retrouver la clé publique de l'autre. Alternativement, elles peuvent échanger directement leurs clés publiques en les envoyant accompagnées de leurs certificats correspondants.
   Un certificat permet de lier une identité à une clé publique. Ce lien est attesté par une autorité de certification dite CA qui fait foi de tiers de confiance. Typiquement, un certificat doit posséder les informations suivantes : identité du détenteur de la clé publique, sa clé publique, validité de la clé publique et enfin signature du tout par le CA. Le standard le plus utilisé pour les certificats est X.509.
3) Authentification : après la distribution des clés publiques, Astrid et Béatrice peuvent s'authentifier. Il y a plusieurs protocoles d'authentification basés sur les clés publiques (notamment celui de Needham-Schroeder). Astrid et Béatrice peuvent échanger des messages de défi-réponse (A et B étant respectivement les identités d'Astrid et de Béatrice):
   i) $A \rightarrow B : \{N_A, A\}_{PK_B}$ ($N_A$ une valeur aléatoire choisie par Astrid)
   ii) $B \rightarrow A : \{N_A, N_B, B\}_{PK_A}$ ($N_B$ une valeur aléatoire choisie par Béatrice)
   iii) $A \rightarrow B : \{N_B\}_{PK_B}$

   L'ajout de l'identité à chaque message de défi empêche les attaques par entrelacement de sessions.
4) Choix de la clé de session : pour pouvoir chiffrer les fichiers d'une manière efficace et rapide, on utilise un algorithme de chiffrement symétrique (ex : AES). Les deux personnes se mettent d'accord sur une clé secrète (clé de session) en utilisant Diffie-Hellman par exemple ou en s'appuyant sur les nonces $N_A$ et $N_B$ déjà échangés entre A et B sous forme chiffré : $K_S=f(N_A, N_B)$.
4) Confidentialité des fichiers : Astrid envoie les fichiers $\{F_i\}$ chiffrés avec la clé de session :
   iv) $A \rightarrow B : \{F_i\}_{K_S}$

## *Exercice 2 :*

1) Avec la technologie actuelle, il est recommandé d'utiliser un algorithme de chiffrement symétrique avec une clé de taille minimale de 64 bits. Puisque DES utilise une clé secrète dont 56 bits sont effectifs, il n'est pas recommandé de chiffrer avec une seule application de DES. En pratique, on utilise 3DES (noté aussi TDES) avec 2 ou 3 clés différentes :
   $3DES = DES_{k_1} \circ DES^{-1}_{k_2} \circ DES_{k_3}$
   Il a été prouvé que 3DES augmente la complexité de cryptanalyse de DES d'un facteur de 2 (c.à.d. 3DES est équivalent à DES avec une clé de 112 bits).
2) $2DES = DES_{k_1} \circ DES_{k_2}$

a) 2DES est plus rapide que 3DES mais plus lent que DES et a priori il est plus robuste que DES. Naïvement, on peut estimer la force cryptographique ajoutée par 2DES à un facteur de 2 puisque on applique deux fois DES avec deux clés différentes.

b) Analyse de sécurité de 2DES :

i) Avec DES, on chiffre un bloc de 64 bits avec une clé de 56 bits. Avec 2DES, on chiffre un bloc de 64 bits avec une clé de 112 bits (56 + 56 bits). Si on considère un message m de 64 bits, le message chiffré c correspondant avec 2DES (c = 2DES(k, m)) est aussi de 64 bits. Pour un message m donné, il y a $2^{64}$ valeurs possibles pour le message chiffré c. Puisqu'il y a $2^{112}$ valeurs possibles pour la clé, en moyenne il y a donc $2^{112}/2^{64} = 2^{48}$ valeurs de clés possibles qui donnent le même message c.

ii) Meet in the middle attack : attaque permettant de faire un compromis entre le temps de calcul et l'espace de stockage utilisé pour la cryptanalyse (Diffie et Hellman 1977).

L'attaquant construit deux listes $L_1$ et $L_2$ telles que :

$$L_1 = \{DES(k, m), \forall k\} \text{ et } L_2=\{DES^{-1}(k, c), \forall k\}$$

Il retrouve ainsi deux listes de textes chiffrés intermédiaires. Parmi ces textes intermédiaires, il y en a en moyenne $2^{48}$ qui sont égaux de part et l'autre des deux listes. L'attaquant retrouve ainsi en moyenne $2^{48}$ clés possibles (meet in the middle). Si l'attaquant connaît une deuxième paire de texte clair/chiffré, il peut refaire les mêmes opérations en utilisant cette fois les clés obtenues à partir de la première paire et ainsi augmenter la probabilité de retrouver la clé secrète. Le nombre de clés possibles en moyenne se réduit à $2^{48}/2^{64} =2^{-16}$. La probabilité que l'attaquant retrouve la clé secrète est donc égale à $1 - 2^{-16}$.

Il faut savoir que pour faire la cryptanalyse de DES, il faut un effort de l'ordre de $2^{55}$ opérations dû à la propriété de complémentarité de DES. Donc, l'attaquant de 2DES effectue en tout $2^{55}$ opérations de chiffrement et $2^{55}$ opérations de déchiffrement pour la première paire de textes clair/chiffré et $2^{48}$ opérations de chiffrement et $2^{48}$ opérations de déchiffrement pour la deuxième paire : un effort de l'ordre de $2\times(2^{55} + 2^{48}) < 2^{57}$ opérations. La cryptanalyse de 2DES nécessite donc un effort de l'ordre d'une attaque de recherche exhaustive d'une clé de 57 bits.

## *Exercice 3*

1) Il faut noter que $31\times89 \equiv 1 \bmod 197$. Donc l'inverse de 89 dans $\mathbb{Z}_{phi}$ avec phi=197 est 31.

2) RSA : Les *clés RSA* sont générées de la façon suivante :
   - Clé publique (n, e) : on choisit deux nombres premiers p et q grands et distincts. On considère le nombre composé n=p.q. On choisit aléatoirement e un entier premier avec phi=(p-1)(q-1).
   - Clé secrète (d, phi) : on calcule phi=(p-1)(q-1) et on dérive un entier d tel que $e.d \equiv 1 \bmod phi$ (c.à.d. : d est l'inverse de e dans $\mathbb{Z}_{phi}$).

   *Chiffrement:* on convertie le message en un entier m | 0 < m < n et on calcule le message chiffré : $c = m^e \bmod n$.

   *Déchiffrement :* pour déchiffrer un message c on calcule : $m' = c^d \bmod n$.

   On obtient m'= m parce que (en utilisant le théorème de Fermat):

   $m' = c^d \bmod n \equiv m^{ed} \bmod n \equiv m^{1+A.phi} \bmod n \equiv m \,.\, (m^{phi})^A \bmod n \equiv m \,.\, 1 \bmod n = m.$

3) e = 3, d = 334, n = 59×17 = 3001

   $c = 4^3 \bmod 3001 = 64$. Connaissant 64, il est facile de retrouver $m=c^{1/e}$ (il faut que m soit $>15 \approx n^{1/e}$). Pour éviter ce problème, on utilise une méthode de remplissage ("padding") : on ajoute au texte clair avant de le chiffrer une valeur structurée et aléatoire. Il y a plusieurs méthodes de remplissage sûres pour RSA ex: OAEP, PKCS #1 (nouvelle version).

4) On note que n=65=5×13, donc p=5, q=13, et phi = (p-1)(q-1) = 4×12 = 48

   Il faut choisir un couple de clés publique/privée qui vérifie $e.d \equiv 1 \bmod 48$. On obtient la table des couples possibles (il faut éliminer les multiples de 2 et 3 qui ne sont pas premiers avec phi):

| e | d |
|---|---|
| **5** | 29 |
| **7** | 7 |
| **11** | 35 |
| **13** | 37 |
| **17** | 17 |
| **19** | 43 |
| **23** | 23 |
| **25** | 25 |
| **31** | 31 |
| **41** | 41 |
| **47** | 47 |

Il y a 4 paires possibles, le couple (e, d) ∈ {(5, 29), (11, 35), (13, 37), (19, 43)}.
Le message chiffré de m=47 est $c=4^5 \bmod 65 = 49 \bmod 65$.
La clé privée associée à e=5 est d=29. Pour faciliter le calcul, on peut faire une exponentiation modulaire.
On sait que $29 = 2^4 + 2^3 + 2^2 + 2^0$, donc on calcule :
$49^2 = 2401 \bmod 65 = 61$
$49^4 = 3721 \bmod 65 = 16$
$49^8 = 256 \bmod 65 = 61$
$49^{16} = 16$
$49^{29} = 16 \times 61 \times 16 \times 49 \bmod 65 = 4$.