Examen Final : Principe et Algorithme Cryptographiques

durée : 1 h 30

Université de Lille-1
FIL — PAC

16 mai 2013

Ce sujet est composé de 4 pages.

1 QCM à auto-correction cryptographique

Cet exercice contient un QCM que vous allez remplir, et que vous allez nous aider à corriger vous-même après y avoir répondu. Le principe est simple : chaque question possède deux réponses possible. A chaque réponse est associé un nombre entier (en **[gras]** à côté des cases). Cochez les bonnes réponses, et inscrivez à la fin la somme des entiers correspondants à vos réponses. Si vous ne connaissez pas la réponse, répondez au hasard. Un programme déduira votre note de la somme que vous allez écrire.

Les mauvaises réponses ne sont pas pénalisés (elles rapportent zéro points). Les bonnes réponses rapportent un point. Comme en répondant au hasard on peut avoir la moitié des points en moyenne, alors si on dit que B est le nombre de bonnes réponses que vous allez cocher, la note que vous aurez à cet exercice est :

$$note_{QCM} = \max(0, B - 17)$$

Le système utilise une technique cryptographique *probablement* résistante à vos éventuelles tentatives de fraude.

1	Avez-vous bien compris le fonctionnement de ce QCM ? ☐ **[6098]** oui ☐ **[7165]** non
2	L'AES est-il un mécanisme de chiffrement à clef publique ? ☐ **[8025]** oui ☐ **[9168]** non
3	Les algorithmes de chiffrement à clef publique sont généralement... ☐ **[4399]** plus lent que ceux à clef secrète ☐ **[3050]** plus rapides
4	Est-ce que 512 bits est une taille de module suffisante pour RSA en 2013 ? ☐ **[1796]** oui ☐ **[5561]** non
5	Si on sait calculer des préimages sur une fonction de hachage, est-ce qu'on sait aussi calculer des secondes préimages ? ☐ **[5477]** oui ☐ **[1712]** non
6	L'AES est-il sûr en 2013 ? ☐ **[6103]** oui ☐ **[7376]** non
7	Est-il possible d'effectuer en pratique une recherche exhaustive sur le DES ? ☐ **[2692]** oui ☐ **[1343]** non
8	Et sur l'AES ? ☐ **[8012]** oui ☐ **[4529]** non
9	Y a-t-il une attaque par le milieu sur le double-DES ? ☐ **[5119]** oui ☐ **[3770]** non
10	En 2013, une fonction de hachage de 128 bits est-elle suffisante ? ☐ **[2094]** oui ☐ **[5653]** non
11	Pour calculer $a^k \bmod p$, l'exponentiation rapide nécessite... ☐ **[3544]** $\mathcal{O}(\log a)$ multiplication ☐ **[61]** $\mathcal{O}(\log k)$ multiplication
12	4^{11} modulo 17 est égal à... ☐ **[8272]** 13 ☐ **[6923]** 15
13	Pour calculer les inverses modulo p, on utilise l'algorithme... ☐ **[1631]** d'Euclide étendu ☐ **[2698]** de Pythagore étendu
14	L'inverse de 5 modulo 17 est... ☐ **[597]** 7 ☐ **[4080]** 9

15	Le PGCD de 330 et 231 est... ☐ **[2276]** 7 ☐ **[6041]** 33
16	Est-il facile de fabriquer de grands nombres premiers aléatoires ? ☐ **[6210]** oui ☐ **[2445]** non
17	Lorsque $n = pq$, la fonction $\phi(n) = (p-1)(q-1)$ donne... ☐ **[2069]** le nombre d'entiers inférieurs à n et qui ont un diviseur commun avec n ☐ **[1002]** le nombre d'entiers inférieurs à n et premiers avec n ?
18	Quel que soit l'entier $a \neq 0$, on a... ☐ **[456]** $a^{\phi(n)} = 0 \bmod n$ ☐ **[1599]** $a^{\phi(n)} = 1 \bmod n$
19	Dans RSA, connaissant n, calculer $\phi(n)$ est aussi dur que calculer la factorisation $n = pq$? ☐ **[2968]** oui ☐ **[1825]** non
20	Dans RSA, connaissant n, calculer l'exposant secret d est aussi dur que calculer la factorisation $n = pq$? ☐ **[5635]** oui ☐ **[2076]** non
21	Dans RSA, un pirate qui saurait déchiffrer sans connaître la clef secrète pourrait reconstituer facilement la clef secrète ? ☐ **[4480]** oui ☐ **[5623]** non
22	Dans RSA, avec un module n, est-il exact que le chiffré des produits est le produit des chiffrés modulo n ? ☐ **[5040]** oui ☐ **[1481]** non
23	Dans RSA, peut-on utiliser $e = 3$ avec un module n de quelques milliers de bits sans prendre de précaution particulière ? ☐ **[4567]** oui ☐ **[5916]** non
24	Calculer le logarithme dans $\mathbb{Z}$ (les entiers) est facile ☐ **[7624]** oui ☐ **[6275]** non
25	Calculer le logarithme dans $\mathbb{Z}/p\mathbb{Z}$ (les entiers modulo p) est facile ☐ **[3354]** oui ☐ **[2287]** non
26	Alice et Bob disposent d'un canal de communication non-fiable. La technique de Diffie-Hellman leur permet... ☐ **[4729]** d'établir un secret commun ☐ **[5796]** de vérifier mutuellement leur identité
27	Si le logarithme discret était facile, est-ce que la technique de Diffie-Helman resterait utilisable ? ☐ **[2793]** oui ☐ **[1726]** non
28	Une bonne fonction de hachage doit-elle être facilement inversible ? ☐ **[9129]** oui ☐ **[5646]** non
29	Le mode opératoire de Merkle-Damgård "transmet" la résistance de la fonction de compressions aux... ☐ **[3827]** collisions ☐ **[2478]** secondes préimages
30	Quand on utilise un MAC, faut-il rendre publique la clef utilisée ? ☐ **[3311]** oui ☐ **[9492]** non
31	Si je signe (electroniquement) un document, est-ce que je peux par la suite prétendre qu'on a imité ma signature ? ☐ **[5553]** oui ☐ **[4486]** non
32	On applique généralement les algorithmes de signatures sur des empreintes des documents à signer car... ☐ **[1002]** Cela produit des signatures plus compactes ☐ **[4485]** cela augmente la sécurité
33	Pour pouvoir vérifier un mot de passe, il vaut mieux stocker... ☐ **[4831]** la paire $(K, AES_K(password))$ ☐ **[8390]** une empreinte du mot de passe
34	Un certificat sert à... ☐ **[6529]** prouver qu'une clef publique appartient bien à une personne donnée ☐ **[348]** prouver le lien entre une clef publique et une clef secrète

Somme des entiers associés à vos réponses :

2 Protocole de Needham-Schroeder

Le protocole de Needham-Schroeder est un protocole qui date de 1978 ; Alice et Bob l'exécutent pour s'assurer de leurs identités respectives (Alice veut être sûre qu'elle parle à Bob, et Bob veut être sûr qu'il parle à Alice). Alice et Bob utilisent tous les deux une méthode de chiffrement à clef publique, et ils possèdent chacun la clef publique de l'autre. On note PK_a et PK_b les clefs publiques. On note $\mathcal{E}_{PK_a}(x)$ le chiffrement de x par la clef publique d'Alice. Pendant l'exécution du protocole, les participants doivent choisir des nombres aléatoires secrets (connus d'eux seuls), qu'on note N_a ou N_b, selon qu'ils appartiennent à Alice ou à Bob. Enfin, on écrit :

$$A \longrightarrow B \quad : \quad x,$$

pour dire qu'Alice envoie le message x à Bob. L'un des deux participants (ici Alice) lance le protocole, qui fonctionne en trois temps :

$$\begin{array}{llcll} 1. & \text{A} & \longrightarrow & \text{B} & : \quad \mathcal{E}_{PK_b}(N_a) \\ 2. & \text{B} & \longrightarrow & \text{A} & : \quad \mathcal{E}_{PK_a}(N_a, N_b) \\ 3. & \text{A} & \longrightarrow & \text{B} & : \quad \mathcal{E}_{PK_b}(N_b) \end{array}$$

1. Expliquez de façon détaillée ce que Bob doit faire pour mener à bien l'étape 2.
2. Que doit vérifier Alice à la fin de l'étape 2 ?
3. Que doit vérifier Bob à la fin de l'étape 3 ?
4. Que se passe-t-il si Edouard (the *Evil guy*) essaye de se faire passer pour Alice auprès de Bob ?

Attaque WITM (Woman-In-The-Middle) Alors que le protocole paraissait sûr, Gavin Lowe a trouvé en 1995 une faille en utilisant un programme de recherche automatique d'attaques qu'il avait conçu.

On suppose maintenant qu'il y a un troisième participant, Irène (the *Intruder*). Irène va tirer parti du fait qu'Alice initie le protocole avec elle pour se faire passer pour Alice auprès de Bob. On note "I(A)" quand Irène essaie de se faire passer pour Alice.

$$\begin{array}{llcrl} \text{1-a.} & \text{A} & \longrightarrow & \text{I} & : \quad \mathcal{E}_{PK_i}(N_a) \\ \text{1-b.} & \text{I(A)} & \longrightarrow & \text{B} & : \quad \mathcal{E}_{PK_b}(N_a) \\ \text{2-b.} & \text{B} & \longrightarrow & \text{I(A)} & : \quad \mathcal{E}_{PK_a}(N_a, N_b) \\ \text{2-a.} & \text{I} & \longrightarrow & \text{A} & : \quad \mathcal{E}_{PK_a}(N_a, N_b) \\ \text{3-a.} & \text{A} & \longrightarrow & \ldots & : \quad \ldots \\ \text{3-b.} & \ldots & \longrightarrow & \ldots & : \quad \ldots \end{array}$$

5. Complétez les deux dernières étapes de l'attaque, et justifiez qu'Irène arrive à convaincre Bob qu'elle est Alice.

Réparation On modifie le protocole de la façon suivante : chaque fois qu'un participant envoie un contenu chiffré, il ajoute son identité à l'intérieur du message chiffré. Le protocole devient donc :

$$\begin{array}{llcll} 1. & \text{A} & \longrightarrow & \text{B} & : \quad \mathcal{E}_{PK_b}(A, N_a) \\ 2. & \text{B} & \longrightarrow & \text{A} & : \quad \mathcal{E}_{PK_a}(B, N_a, N_b) \\ 3. & \text{A} & \longrightarrow & \text{B} & : \quad \mathcal{E}_{PK_b}(A, N_b) \end{array}$$

On suppose aussi que les participants au protocole vérifient que l'identité supposée des participants correspond à celle qui figure dans les message.

6. Ceci empêche-t-il l'attaque par le milieu ?

3 Fonction de hachage de Chaum, van Heijst et Pfitzmann

Préliminaires mathématiques. Cette section ne contient pas de question, mais vous devez la lire attentivement avant de passer à la suite.

Soit p un nombre premier. On rappelle qu'un élément x de $\mathbb{Z}/p\mathbb{Z}^*$ est un générateur si tous les éléments de $\mathbb{Z}/p\mathbb{Z}^*$ sont des puissances de x (modulo p). Dans ce cas, si $x^i = x^j \bmod p$, alors $i = j \bmod p-1$.

L'exercice utilise les deux résultats suivants, mais leur utilisation sera signalée au bon moment.

Lemme 1 (Lemme de Gauss)**.** *Si a divise bc et que a est premier avec b, alors a divise c.*

Description de la fonction. Considérons un (grand) nombre premier q, choisi de telle sorte que $p = 2q + 1$ est premier lui aussi. On prend aussi deux générateurs de α et β du groupe $\mathbb{Z}/p\mathbb{Z}^*$. La fonction de hachage prend en entrée deux nombres x et y dans l'intervalle $\{0, \ldots, q-1\}$ et calcule :

$$H(x, y) = \alpha^x \beta^y \bmod p$$

1. Supposons que q s'écrive sur n bits. Quelle sont les tailles (en bits) de l'entrée de H et de sa sortie ?
2. Quelle est (en fonction de n) la complexité asymptotique du calcul de H ?
3. Comment faire pour hacher des messages plus long sans augmenter la taille de q ?
4. Comme on a supposé que α était un élément primitif, alors on sait qu'il existe λ tel que $\beta = \alpha^\lambda \bmod p$. Si α et β nous sont imposés, déterminer λ revient à résoudre un problème algorithme usuel en cryptologie à clef publique. Lequel ?
5. Ce problème est-il facile à résoudre, pour de grandes valeurs de p ?
6. Montrez que si on connaît la valeur de λ, alors on peut facilement forger des collisions pour la fonction.

Résistance aux collisions On a donc vu que si on connaît le logarithme discret de β en base α, alors on peut fabriquer des collisions. L'objectif des questions suivantes est de montrer la réciproque : si on arrive à trouver *une seule* collision, on est capable de calculer le logarithme discret de β.

7. Expliquez pourquoi cela (la "réciproque" évoquée ci-dessus) garantit la résistance aux collisions de la fonction de hachage
8. Supposons qu'on ait une collision sur H, c'est-à-dire deux paires $(x, y) \neq (u, v)$ telles que $H(x, y) = H(u, v)$. Justifiez qu'on a alors :
$$\alpha^{x-u} = \beta^{v-y} \bmod p$$
9. Justifiez qu'on a :
$$\lambda(v - y) - (x - u) = 0 \bmod p - 1$$

Les questions suivantes sont plus difficiles et plus "mathématiques".

10. On pose $d = \mathrm{PGCD}(z - y, p - 1)$. Rappelons que c'est le plus grand entier (positif) qui divise à la fois $z - y$ et $p - 1$. En utilisant le fait que z et y sont strictements inférieurs à q, montrez que d est strictement inférieur à q (on peut supposer que $z - y > 0$).
11. En utilisant le fait que d divise $p - 1$ et le lemme de Gauss, justifier que d vaut soit 1, soit 2.
12. Dans le cas où $d = 1$, justifiez qu'il y a une seule valeur de λ possible, et trouvez comment on peut la calculer facilement.

Le cas où $d = 2$ est plus pénible, car l'équation en λ a deux solutions, et il est plus difficile de les calculer (essayez à la maison !). C'est néanmoins possible, et on peut tester laquelle des deux solutions réalise effectivement $\beta = \alpha^\lambda$. On peut donc calculer le log de β dans tous les cas.